2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

Low-Drift Visual Odometry in Structured Environments by Decoupling Rotational and Translational Motion

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

***Abstract*— We present a low-drift visual odometry algorithm that separately estimates rotational and translational motion from lines, planes, and points found in RGB-D images. Previous methods estimate drift-free rotational motion from structural regularities to reduce drift in the rotation estimate, which is the primary source of positioning inaccuracy in visual odometry. However, multiple orthogonal planes are required to be visible throughout the entire motion estimation process; otherwise, these VO approaches fail. We propose a new approach to estimate drift-free rotational motion jointly from both lines and planes by exploiting environmental regularities. We track the spatial regularities with an efficient $\mathbf{SO}(3)$-manifold constrained mean shift algorithm. Once the drift-free rotation is found, we recover the translational motion from all tracked points with and without depth by minimizing the de-rotated reprojection error. We compare the proposed algorithm to other state-of-the-art visual odometry methods on a variety of RGB-D datasets (including especially challenging pure rotations) and demonstrate improved accuracy and lower drift error.**

I. Introduction

Visual odometry (VO) algorithms estimate the six degrees of freedom (DoF) rotational and translational camera motion from a sequence of images. They are a fundamental tool for applications from augmented reality to autonomous robots.

Many VO and Visual Simultaneous Localization and Mapping (V-SLAM) approaches, which jointly estimate rotational and translational motion, have shown promising results. However, these approaches cannot avoid drift in the rotation estimate without SLAM techniques (loop closure, global 3D map construction), resulting in large drift errors because the main source of positional inaccuracy in VO is rotation estimation error [1], [2], [3]. Many visual navigation methods are also unstable for pure, on the spot rotations [4], [5].

Our previous work [6] introduced *Orthogonal Plane-based Visual Odometry* (OPVO) to address these issues. OPVO exploits orthogonal planar structures to determine the absolute, drift-free orientation of an RGB-D camera. Based on the absolute camera orientation, it finds the optimal translation by minimizing the de-rotated reprojection error from tracked points with depth information. Although OPVO drastically reduces the drift error, there are still two key limitations: OPVO requires at least two orthogonal planes to be visible at all times, and point features with depth information. Also, the experimental evaluation in [6] was performed mainly on synthetic RGB-D datasets.

[1]P. Kim and H.J. Kim are with Department of Mechanical and Aerospace Engineering, IAAT, Seoul National University, Seoul, South Korea. {rlavywls,hjinkim}@snu.ac.kr

[2]B. Coltin is with SGT, Inc., NASA Ames Research Center, Moffett Field, CA, 94035, USA. {brian.j.coltin}@nasa.gov

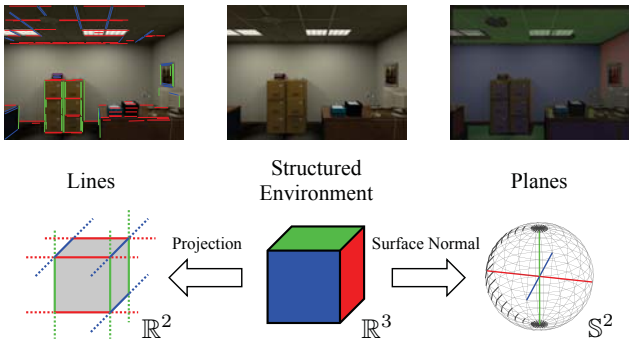

Fig. 1. Example of a structured environment exhibiting strong orthogonal spatial regularities. For drift-free and stable rotation estimation, we exploit line and plane primitives together to recognize the structural regularities with respect to an RGB-D camera.

To address these issues, we propose *Line and Plane based Visual Odometry* (LPVO), a novel VO algorithm that exploits line and plane primitives jointly to recognize the spatial regularities of orthogonal structured environments (see Fig. 1). Lines from RGB images and surface normal vectors from depth images are simultaneously used to perceive environmental regularities accurately and stably. LPVO can track drift-free rotational motion while at least a single plane and a pair of lines parallel to the Manhattan world (MW) axes are visible. Furthermore, we utilize point features without depth information when we recover the optimal translational motion. Extensive evaluations show that LPVO produces the lowest drift error compared to other state-of-the-art VO methods, including OPVO [6]. The main contributions of this paper are:

- We propose a novel approach to estimate absolute and drift-free rotational motion jointly from both lines and planes by utilizing environmental regularities.
- We newly use tracked points with no depth information to recover the 3-DoF translation.
- We evaluate the VO algorithms on the TUM [7] and TAMU [8] RGB-D datasets, as well as a new dataset traversing a large building, showing low drift for LPVO.

II. Related Work

VO and V-SLAM methods are being actively researched to lower the rate of VO drift. From the vast literature in VO and V-SLAM, we review a subset of state-of-the-art contributions, existing VO methods with motion decoupling, and studies specifically focusing on accurate rotation estimation.

VO algorithms can be classified into indirect, direct, and hybrid methods depending on the type of visual information

978-1-5386-3081-5/18/$31.00 ©2018 IEEE

used [4]. The most widely used indirect methods, point feature-based methods, have proven successful for 6-DoF motion estimation [9], [10], [11]. In [10], low drift error is achieved using salient feature points both with and without depth information. The recent ORB-SLAM2 [11] shows outstanding motion estimation performance with monocular, stereo, and RGB-D cameras using the same ORB features for all SLAM tasks. To reduce drift error, however, ORB-SLAM2 relies heavily on SLAM techniques (loop closing, relocalization, local 3D map reuse), which require substantial memory and computation. Direct VO methods [12], [13], [4] estimate 6-DoF camera motion by minimizing the photometric error between image frames. But they suffer from drift caused by unmodeled visual effects such as irregular illumination changes, and fare poorly at tracking on-the-spot rotations.

Some research has estimated rotational and translational motion separately. Rotation is estimated using epipolar geometry, and the translation is recovered with triangulated 3D points [14]. [15] splits camera motion into the separate rotation and translation estimates using distant and close points with a disparity map and the camera speed, while [16] estimates rotation and translation separately with carefully selected features. These VO methods, however, cannot estimate drift-free rotation in structured environments because it is difficult to recognize environmental regularities using point primitives. To utilize structural information, [17] detects dominant bundles of parallel lines for rotation and estimates translation from a 2-point algorithm up to a scale. OPVO [6] tracks a Manhattan frame (MF) for absolute camera orientation from surface normal vectors, and recovers translation by minimizing de-rotated reprojection error with available depth points. Although these approaches use structural features, no existing approach uses both lines and planes.

Several studies have more focused on accurate rotation estimation in structured environments due to the importance of rotational motion [18]. From the line segments in the image, [19] estimates the rotational motion by finding orthogonal vanishing points (VPs) with a 3-line RANSAC algorithm. While this method can estimate drift-free rotation using RGB images only, the performance is sensitive to the quality of visible lines. [2] derives MF inference algorithms based on the distribution of the surface normal vectors from depth images. In [20] and [6], drift-free rotation estimation is performed with a mean-shift algorithm based on the surface normal vector distribution. While these methods demonstrate superior rotation estimation in structured environments, at least two orthogonal planes must always be visible.

III. Background 3D Geometry

A. Gaussian Sphere

A Gaussian sphere is a unit sphere centered on the center of projection (COP) of a camera, and is a convenient method to represent geometric elements such as lines and normal vectors when the camera intrinsic parameters are known. A line in the image is projected onto the Gaussian sphere as a great circle (the intersection of the unit sphere and the plane defined by the line and the COP, see Fig. 2). The great circle of each line can be expressed as a unit vector in the Gaussian sphere. Great circles representing parallel lines in the image intersect at two antipodal points on the Gaussian sphere. A unit vector from the COP to the intersection point is a vanishing direction (VD), calculated as the cross product of the normal vectors of two great circles representing parallel lines in the image. The three orthogonal VDs defined by the parallel lines match the orthogonal surface normal vectors of the planes in a perfect Manhattan world. These vectors form the basis of a Manhattan frame.

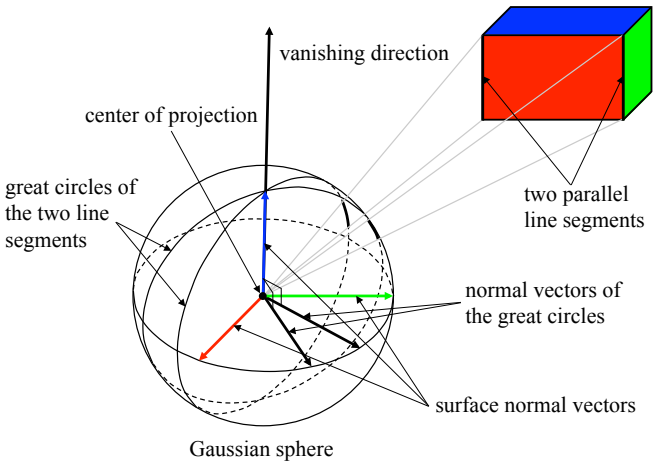

Fig. 2. Background 3D geometric relationship between the lines, planes, and the Gaussian sphere with an orthogonal structure. A vanishing direction vector is defined by at least two parallel line segments. Each orthogonal plane and its corresponding normal vector are drawn with the same color.

B. Rotation Motion with Vanishing Directions

In Euclidean 3D space, we represent the 6-DoF camera motion as the 4×4 rigid body transformation matrix $T \in \mathrm{SE}(3)$, composed of the 3-DoF rotational motion $R \in \mathrm{SO}(3)$ and the 3-DoF translational motion $\mathbf{t} \in \mathbb{R}^3$. A vanishing direction $\mathbf{d} \in \mathbb{R}^3$ on the Gaussian sphere can be transformed into $\mathbf{d}'$ by the 6-DoF camera motion as:

$$\tilde{\mathbf{d}}' = T\tilde{\mathbf{d}} = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ 0 \end{bmatrix} = \begin{bmatrix} R\mathbf{d} \\ 0 \end{bmatrix} \tag{1}$$

where $\tilde{\mathbf{d}} = \begin{bmatrix} \mathbf{d}^\top & 0 \end{bmatrix}^\top \in \mathbb{P}^3$ denotes the VD in homogeneous coordinates. From Eq. (1), we can observe that the VD is only dependent on the rotational motion of the camera.

IV. Proposed Method

A. Orthogonal Plane-based Visual Odometry

The proposed *Line and Plane based Visual Odometry* (LPVO) method builds on our previous *Orthogonal Plane-based Visual Odometry* (OPVO) algorithm [6], which we summarize briefly (for full details, refer to [6]). OPVO has two main steps: 1) structural regularities (Manhattan frame) are tracked to estimate drift-free rotation with a $\mathrm{SO}(3)$-manifold constrained mean shift algorithm; and 2) translational motion is recovered by minimizing the de-rotated reprojection error from tracked points.

The core of the OPVO rotation estimation is tracking the Manhattan frame with a SO(3)-manifold constrained mean shift algorithm based on the tangent space Gaussian MF (TG-MF) model [21] under the assumption that the MF does not change too much between the frame-to-frame motion. Given the density distribution of surface normal vectors on the Gaussian sphere $\mathbb{S}^2$, OPVO infers the mean of the surface normal 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 algorithm are projected onto the SO(3) manifold to maintain orthogonality, resulting in the absolute orientation estimate of the camera.

For the translation estimation, OPVO transforms feature correspondences between consecutive frames into a pure translation by taking advantage of the drift-free rotation estimation in the previous step. OPVO recovers the 3-DoF translational motion of the camera by minimizing de-rotated reprojection error from the tracked points, which is only dependent on the translational movement.

Next, we present LPVO, a new approach to exploit line and plane information jointly for stable and accurate drift-free rotation estimation even when only a single plane is visible. For more accurate translation estimation, we additionally use tracked points without depth information. Fig. 4 shows an overview of the LPVO algorithm.

B. Drift-Free Rotation Estimation with Lines and Planes

We extract the vanishing directions from lines in the RGB images and surface normal vectors from planes in the depth images to determine the camera orientation relative to the Manhattan frame. To extract the vanishing direction vectors [17], line features over a fixed length (in our experiments, 25 pixels) are detected using LSD [22]. Given the N detected line features, we compute their corresponding great circle unit normal vectors. From the N associated normal vectors, we calculate $\binom{N}{2}$ vanishing directions (one for every possible pair of lines) by taking the cross product of each pair of normal vectors.

We also extract surface normal vectors for every pixel point from the depth image with an RGB-D camera [23]. We pre-process the depth image with a simple box filter to remove noise in the raw depth data. Unit surface normal vectors on the Gaussian sphere $\mathbb{S}^2$ are computed by the cross product of two tangential vectors, which are tangential to the local surface at the 3D points in the point cloud. In order to remove noise from the tangential vectors, we average the surrounding tangential vectors within a certain neighborhood, which can be done efficiently and quickly using integral images. For further details, see [23].

We represent the extracted vanishing directions and surface normal vectors as 3D points on the concentric spheres $\mathbb{S}^2$ in Fig. 3. Purple points on the inner sphere denote the VDs from lines, and grey points on the outer sphere are the surface normal vectors from planes, showing that the two types of directional vectors from lines and planes gather together around the Manhattan frame axes. The number of

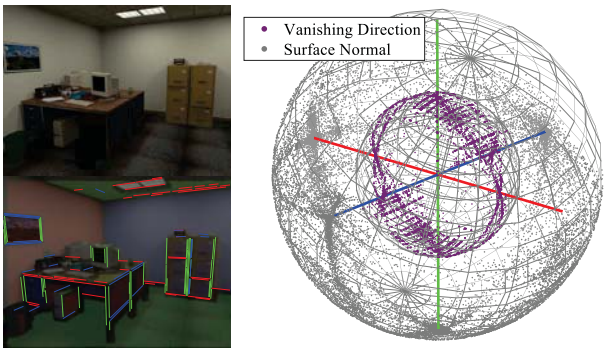

Fig. 3. Clustered lines and segmented planes with inferred MF (left-bottom) are overlaid on the original RGB image (left-top). We employ the concentric spheres (right) with different radius to describe the density distribution of the VDs (purple) and SNs (grey) effectively as 3D points. We can observe that the directional vectors on the both inner (the VDs from lines) and outer (the SNs from planes) sphere are distributed together around the MF axes.

the VDs and surface normals is constantly changing by various factors such as the number of detected lines, invalid per-pixel depth, and an environmental condition. Although most of the directional vectors in MW are distributed around the basis axes of the Manhattan frame, some other points are not near the Manhattan frame because the real 3D world is not a perfect and noise-free Manhattan world. Therefore, it is very important to extract accurate and reliable VDs and surface normal vectors since the density distribution of the directional vectors directly affects the accuracy of rotational motion estimation. LPVO enables MF tracking even when viewing only a single plane with the help of the lines, unlike the previous MF tracking methods [20], [21], [6].

C. Translation Estimation with All Tracked Points

We detect and track the feature points with the Good Features to Track [24] and KLT tracker [25] (for further details, see [6]). We recover the 3-DoF translational motion of the camera by minimizing the de-rotated reprojection error based on tracked points with and without depth information, which is only dependent on the translational movement. We start with the mathematical relationship between the frame-to-frame 6-DoF camera motion and the i-th tracked point feature [10]:

$$\begin{aligned} \mathbf{X}_i^k = Z_i^k \bar{\mathbf{X}}_i^k &= R\mathbf{X}_i^{k-1} + \mathbf{t} \\ &= Z_i^{k-1} R \bar{\mathbf{X}}_i^{k-1} + \mathbf{t} \end{aligned} \tag{2}$$

where $\mathbf{X}_i^k = [X_i^k, Y_i^k, Z_i^k]^\top$ is the 3D coordinates of the point feature in the camera frame at time k, and $\bar{\mathbf{X}}_i^k = [\bar{X}_i^k, \bar{Y}_i^k, 1]^\top$ is the normalized $\mathbf{X}_i^k$ divided by the depth. The rotation R and the translation $\mathbf{t}$ form a rigid body transformation as explained in Section III-B. For a feature with known depth at time $k-1$, we derive two constraint equations from the first row of Eq. (2) by substituting Z_i^k in the third row into the first and second rows, respectively:

$$\begin{aligned} r_{i_1}(\mathbf{t}) &= (R_1 - \bar{X}_i^k R_3)\mathbf{X}_i^{k-1} + \mathbf{t}_1 - \bar{X}_i^k \mathbf{t}_3 = 0 \\ r_{i_2}(\mathbf{t}) &= (R_2 - \bar{Y}_i^k R_3)\mathbf{X}_i^{k-1} + \mathbf{t}_2 - \bar{Y}_i^k \mathbf{t}_3 = 0 \end{aligned} \tag{3}$$

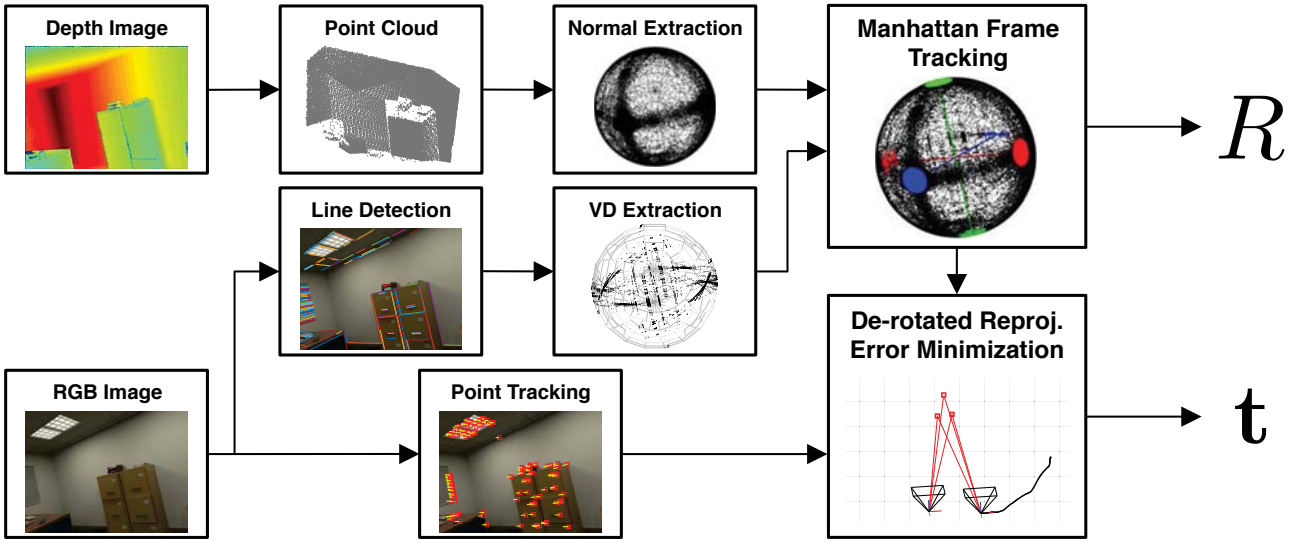

Fig. 4. Overview of LPVO. We estimate the camera rotation (R) by tracking the MF from the vanishing directions and surface normals. Once the absolute orientation is known, we recover translational motion ($\mathbf{t}$) by minimizing a de-rotated reprojection error based on tracked points with and without depth.

where R_h and $\mathbf{t}_h$, $h \in \{1, 2, 3\}$ are h-th rows of R and $\mathbf{t}$ respectively. For a feature with unknown depth, we can derive one constraint equation from the second row of Eq. (2) by combining all rows to eliminate both Z_i^k and Z_i^{k-1}:

$$r'_i(\mathbf{t}) = \mathbf{p} R \bar{\mathbf{X}}_i^{k-1} = 0 \tag{4}$$

$$\text{where } \mathbf{p} = \begin{bmatrix} -\bar{Y}_i^k \mathbf{t}_3 + \mathbf{t}_2 & \bar{X}_i^k \mathbf{t}_3 - \mathbf{t}_1 & -\bar{X}_i^k \mathbf{t}_2 + \bar{Y}_i^k \mathbf{t}_1 \end{bmatrix}$$

There are two residual equations for features with depth, and one residual equation for those without depth. Since we already estimated the drift-free rotational motion R, the residual terms in Eqs. (3) and (4) are only a function of the translational camera motion $\mathbf{t}$. The optimal 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:

$$\mathbf{t}^* = \arg\min_{\mathbf{t}} \sum_{i=1}^{M} \left(r_{i_1}(\mathbf{t})\right)^2 + \left(r_{i_2}(\mathbf{t})\right)^2 + \sum_{i=1}^{N} \left(r'_i(\mathbf{t})\right)^2 \tag{5}$$

where M and N are the number of tracked features with known and unknown depth, respectively. We use the Levenberg–Marquardt (LM) algorithm for solving Eq. (5). By additionally constraining the 3-DoF translation from tracked points without depth, we can estimate more accurate translational motion compared to our previous approach. Note that the proposed method is less sensitive to the existence of enough textures and brightness in the image than the typical feature-based VO methods [11], [10] since the minimum number of points for estimating translation only is smaller than the number of feature points to determine both rotation and translation.

V. Evaluation

We evaluate LPVO on a variety of RGB-D datasets in man-made structured environments:

- *ICL-NUIM* [26] is a synthetic dataset consisting of a collection of RGB and depth images at 30 Hz captured in a living room and office with ground-truth camera poses. The synthesized RGB and depth images are corrupted by the modeled sensor noise to simulate typically observed real world artifacts. It is challenging to estimate the camera trajectory accurately due to low texture and frequent on-the-spot rotations.
- *TUM RGB-D* [7] is a famous dataset for VO evaluation, containing RGB-D images from a Microsoft Kinect RGB-D camera in various indoor environments. It is recorded in room-scale environments with ground-truth trajectories provided by a motion capture system.
- *TAMU RGB-D* [8] contains RGB-D images at 30 Hz recorded in larger scale man-made environments like corridors and stairs inside a building.
- *Author-collected RGB-D dataset* consists of RGB and depth images at 30 Hz with an Asus Xtion Pro Live RGB-D camera in large building-scale indoor environments over 100 m traveling distance.

We compare the proposed LPVO method against other state-of-the-art VO algorithms, including indirect, direct, and hybrid methods, namely ORB [5], DEMO [10], DVO [13], MWO [20], and OPVO [6]. ORB, DEMO, and DVO estimate the rotational and translational motion jointly, while MWO and OPVO decouple the estimation of the rotational and translational motion like LPVO. Recall that the proposed LPVO builds on our previous work OPVO [6]. We deactivate the capability to detect loop closures via image retrieval in ORB for a fair comparison.

The proposed LPVO written in unoptimized MATLAB codes is able to run at 13.5 Hz on a desktop computer with an Intel Core i5 (3.20 GHz) and 8 GB memory, suggesting potential when implemented in C/C++ in the near future.

TABLE I

EVALUATION RESULTS ON ICL-NUIM BENCHMARK

Experiment	LPVO	ORB	DEMO	DVO	MWO	OPVO	Length (m)
Living Room 0	**0.01**	0.02	0.14	0.22	×	×	4.14
Living Room 1	0.04	**0.03**	0.15	0.07	0.32	0.04	2.05
Living Room 2	**0.03**	0.07	0.62	0.50	0.11	0.06	8.42
Living Room 3	0.10	**0.07**	0.33	0.43	0.40	0.10	5.95
Office Room 0	**0.06**	0.20	0.34	0.37	0.31	0.06	6.53
Office Room 1	**0.05**	0.60	0.37	0.36	1.10	0.05	6.72
Office Room 2	**0.04**	0.30	0.76	0.58	×	×	9.01
Office Room 3	**0.03**	0.46	0.18	0.30	1.38	0.04	7.82

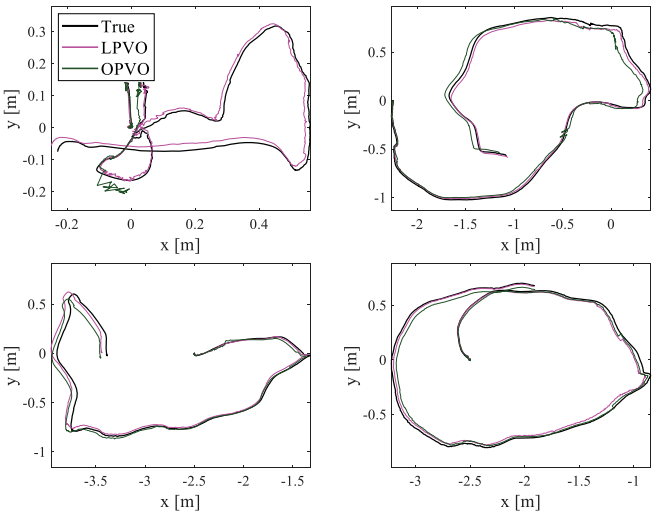

Fig. 5. Estimated trajectories with LPVO (magenta), OPVO (dark green), and ground-truth (black) in the ICL-NUIM dataset Living Room 0, 2 and Office Room 1, 3.

A. ICL-NUIM Dataset

We measure the root mean squared error (RMSE) of the absolute translational error and present the results in Table I. The smallest error for each dataset is bolded. ORB results are from [27]. MWO and OPVO sometimes fail to track the camera (marked as × in Table I) due to multiple orthogonal planes not always being visible. For example, in 'Living Room 0', at one point OPVO sees only a single plane, leading to failure. LPVO can continue estimating the motion stably as shown in the top left of Fig. 5. Our method outperforms the other VO algorithms for most test cases. In two cases, ORB performs better thanks to sufficient texture and local map construction, but the proposed algorithm performs nearly as well. The average translational RMSE of the proposed LPVO is 0.04, while ORB, DEMO, DVO, MWO, and OPVO are 0.21, 0.36, 0.35, 0.60, and 0.06, respectively. The main reason for the improved performance is that LPVO accurately tracks rotations even when the camera rotates in a place by exploiting both lines and surface normal vectors to recognize structural regularities. Although OPVO also estimates accurate camera rotation, it is unstable and fails when only a single plane is visible.

The strength of LPVO becomes clear when plotting the rotation and translation errors for the dataset 'Office Room 3' in Fig. 6. While the rotation error of ORB, DEMO, and DVO

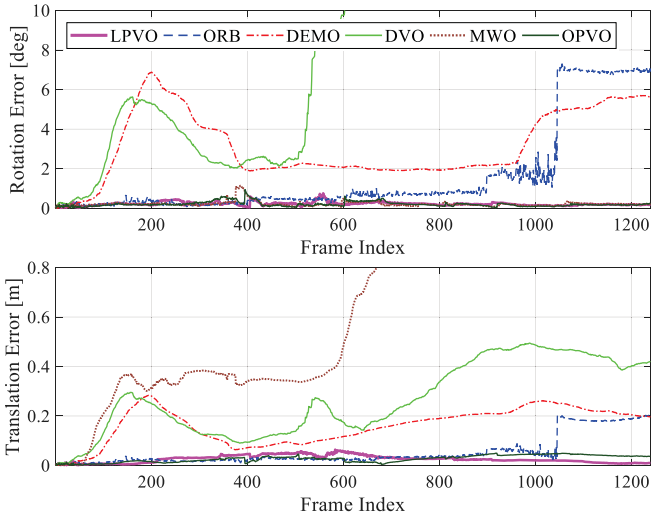

Fig. 6. Absolute rotational error (top) and translational error (bottom) for the proposed and other VO algorithms are plotted. The proposed method shows the lowest rotation and translation error against other VO methods.

TABLE II

EVALUATION RESULTS ON TUM RGB-D BENCHMARK

Experiment	LPVO	ORB	DEMO	DVO	MWO	OPVO	Length (m)
fr3_longoffice	0.19	**0.02**	1.50	0.61	×	×	22.14
fr3_struc_notex_far	**0.07**	0.28	0.40	0.59	0.47	0.13	1.66
fr3_struc_notex_near	**0.08**	0.63	2.59	0.73	0.95	0.16	2.05
fr3_struc_tex_far	0.17	**0.03**	0.06	0.13	1.57	0.18	6.04
fr3_struc_tex_near	0.11	**0.03**	0.20	0.08	0.62	0.19	5.21
fr3_large_cabinet	**0.28**	0.47	0.96	0.97	0.83	0.51	12.37

gradually increase over time, LPVO, MWO, and OPVO drift less than 0.5 degrees thanks to drift-free rotation estimation. The average rotation error of the proposed method is 0.22 degrees whereas ORB, DEMO, DVO, MWO, and OPVO are 1.63, 3.15, 8.12, 0.22, 0.21 degrees respectively. We can also observe that the translational error mainly occurs due to the drift of rotation estimate in the bottom of Fig. 6. Because there are sufficient orthogonal planes, MWO and OPVO can also estimate accurate and drift-free rotational motion. However, LPVO estimates more accurate translational motion by minimizing the de-rotated reprojection error from tracked points with and without depth information.

B. TUM RGB-D Dataset

We evaluate the motion estimation results on a subset of image sequences which contain sufficient structural regularities (lines and planes) in the observed scenes. Table II compares results of the VO methods. Estimated camera trajectories with the ground-truth, LPVO, and OPVO are shown in Fig. 7. We observe that ORB outperforms the proposed algorithm in incomplete (ambiguous) structured environments such as 'fr3_longoffice'. However, LPVO shows better performance in very low texture environments with the help of structural information. LPVO can also work in imperfect structural environments like 'fr3_longoffice' whereas MWO and OPVO require at least two orthogonal planes throughout the entire motion estimation process. When there

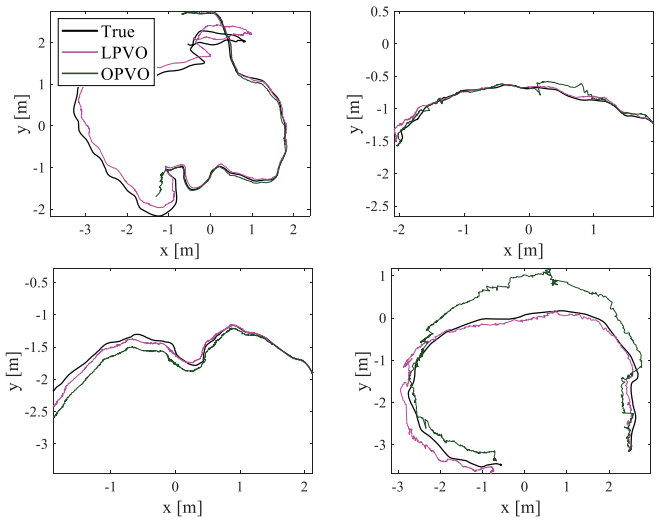

Fig. 7. Estimated trajectories with LPVO (magenta), OPVO (dark green), and ground-truth (black) in the TUM fr3_longoffice, fr3_struc_notex_far, fr3_struc_tex_near, and fr3_large_cabinet.

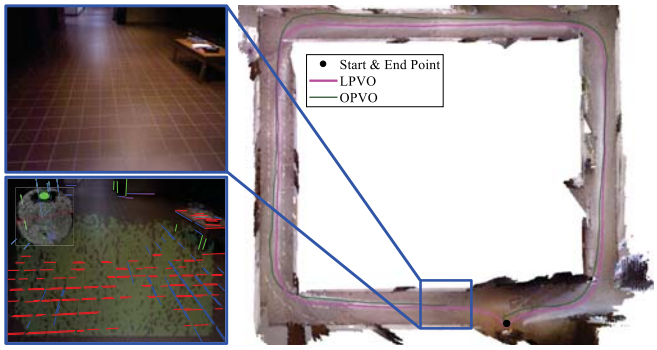

Fig. 8. Example image from 'Corridor-A-const', the clustered lines/planes, and the inferred MF orientation are shown on the left. Since an RGB-D camera looks at only a single plane from this blue box point, OPVO fails to estimate camera motion while LPVO does not. The 3D point cloud is rendered by back-projecting the depth sensor from the estimated camera trajectory with the proposed method. No fusion is performed.

is only a single plane visible, OPVO fails but the proposed method does not as shown on the top left of Fig. 7.

C. TAMU RGB-D Dataset

We present a 3D reconstruction result of 'Corridor-A-const' in the TAMU dataset based on the motion estimation of LPVO in Fig. 8. The trajectory is about 88 meters long, and includes four pure rotational movements, difficult textures, and a segment where the camera looks at only a single plane as shown on the left of Fig. 8. LPVO stably tracks the 6-DoF camera motion even when looking at only a single plane while OPVO fails to estimate rotational motion of a camera, resulting in overall motion estimation failure. Therefore, LPVO accurately estimates the entire camera motion, and achieves final drift error lower than 0.3%, which is the final positioning error divided by the total traveling distance. The start and end points of the estimated camera trajectory accurately meet. The drift-free rotation estimates act like an indoor 3-DoF compass in the long square corridor,

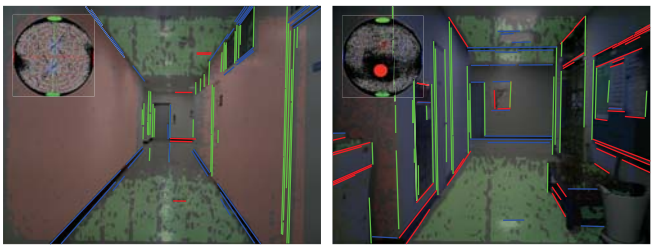

Fig. 9. Example images from the author-collected RGB-D dataset. Inferred MF orientations with clustered lines/planes are overlaid on top of the RGB images from the single-loop (left) and multiple-loop (right) sequences.

resulting in consistent and low-drift motion estimation. Our method preserves the orthogonality of the reconstructed 3D point cloud well, which is rendered by back-projecting the depth image from the estimated camera poses. Note that we do not perform any additional SLAM techniques like 3D local map fusion, loop detection & closure, and relocalization, but the consistent 3D reconstruction result indicates the high accuracy of our VO approach.

D. Author-collected RGB-D Dataset

Finally, we demonstrate that the proposed VO method can work in building-scale indoor environments like long corridors. Fig. 9 shows excerpts from the 'single-loop' (left) and 'multiple-loop' (right) datasets, with trajectory lengths of 93 m and 120 m respectively. The dataset was taken on long square corridors of two different buildings, and is very challenging due to frequent on-the-spot rotations and difficult textures. For evaluating the VO algorithms without a ground-truth trajectory, we collect the dataset on closed-loop trajectories where the starting and end points coincide.

The resulting trajectories for all algorithms are shown in Fig. 10. With LPVO, the starting and ending points nearly match; for the others, they do not. LPVO's final drift error is under 0.2%. LPVO robustly and accurately tracks the 6-DoF camera motion, preserving the orthogonality of the estimated corridor trajectory in the square building.

Similarly, for the 'multiple-loop' dataset (see right of Fig. 10) the start and end points meet only with the proposed algorithm, with final drift error under 0.7%. Although MWO and OPVO can perform drift-free rotation estimation, inaccuracies in translational motion estimation as discussed in Section V-A cause errors to accumulate. The reconstructed trajectory with the proposed method preserves the orthogonality of the corridors in the square building, demonstrating the high quality of the motion estimation. Please refer to the video clips submitted with this paper showing more details about the experiments.

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

VI. Conclusion

We propose a new visual odometry algorithm that is able to perform accurate and low-drift motion estimation in

[1]Video available at `https://youtu.be/mt3kbv2TJZw`

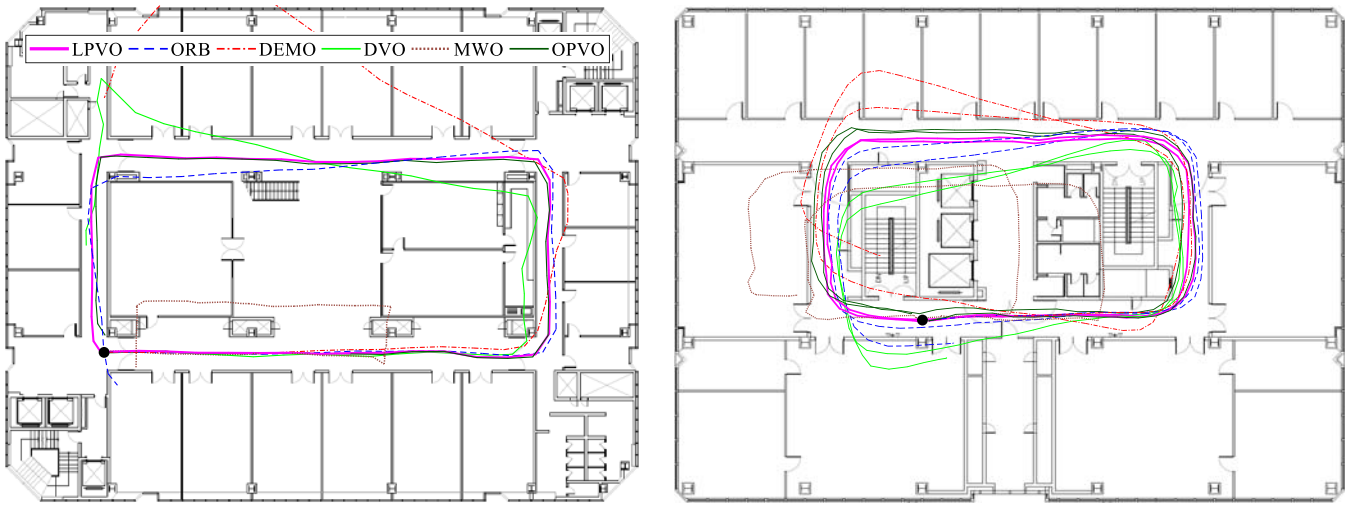

Fig. 10. Estimated trajectories with the proposed and other VO methods on the author-collected dataset in a single-loop (left) and multiple-loop (right) sequences. We start and end at the same point marked in the black circle to check loop closing in the estimated trajectories.

structured environments by decoupling the camera motion into a separate rotation and translation estimation. We newly exploit line and plane primitives together to deal with the degenerate case in the previous drift-free rotation estimation methods, resulting in stable and accurate zero-drift rotation estimation. Given the absolute camera orientation, we recover the optimal translational motion, which minimizes derotated reprojection error based on all tracked points with and without depth. The proposed algorithm is tested thoroughly with a large number of datasets, and shows accurate and low-drift motion estimation results in structural environments. Our method is currently tested with an RGB-D camera in indoor environments. In the future, we will try to implement the proposed algorithm with a stereo camera and possibly extend to outdoor urban environments.

ACKNOWLEDGEMENTS

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science, ICT & Future Planning (2014M1A3A3A02034854) and the Samsung Smart Campus Research Center at Seoul National University (0115-20170013).

REFERENCES

[1] C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimone, "Stereo ego-motion improvements for robust rover navigation," in *IEEE ICRA*, 2001.
[2] J. Straub, N. Bhandari, J. J. Leonard, and J. W. Fisher, "Real-time Manhattan world rotation estimation in 3D," in *IEEE IROS*, 2015.
[3] Y. Zhou, L. Kneip, and H. Li, "Real-time rotation estimation for dense depth sensors in piece-wise planar environments," in *IEEE IROS*, 2016.
[4] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE T-PAMI*, 2018.
[5] R. Mur-Artal, J. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE T-RO*, 2015.
[6] P. Kim, B. Coltin, and H. J. Kim, "Visual odometry with drift-free rotation estimation using indoor scene regularities," in *BMVC*, 2017.
[7] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *IEEE IROS*, 2012.
[8] Y. Lu and D. Song, "Robustness to lighting variations: An RGB-D indoor visual odometry using line segments," in *IEEE IROS*, 2015.
[9] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3D reconstruction in real-time," in *IEEE IV*, 2011.
[10] J. Zhang, M. Kaess, and S. Singh, "A real-time method for depth enhanced visual odometry," *AURO*, 2017.
[11] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE T-RO*, 2017.
[12] A. I. Comport, E. Malis, and P. Rives, "Real-time quadrifocal visual odometry," *IJRR*, 2010.
[13] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for RGB-D cameras," in *IEEE ICRA*, 2013.
[14] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis, "Monocular visual odometry in urban environments using an omnidirectional camera," in *IEEE IROS*, 2008.
[15] M. Kaess, K. Ni, and F. Dellaert, "Flow separation for fast and robust stereo odometry," in *IEEE ICRA*, 2009.
[16] I. Cvišić and I. Petrović, "Stereo odometry based on careful feature selection and tracking," in *IEEE ECMR*, 2015.
[17] J. C. Bazin, C. Demonceaux, P. Vasseur, and I. Kweon, "Motion estimation by decoupling rotation and translation in catadioptric vision," *CVIU*, 2010.
[18] 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 ICRA*, 2015.
[19] J.-C. Bazin and M. Pollefeys, "3-line RANSAC for orthogonal vanishing point detection," in *IEEE IROS*, 2012.
[20] Y. Zhou, L. Kneip, C. Rodriguez, and H. Li, "Divide and conquer: Efficient density-based tracking of 3D sensors in Manhattan worlds," in *ACCV*, 2016.
[21] 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 T-PAMI*, 2017.
[22] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE T-PAMI*, 2010.
[23] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, "Real-time plane segmentation using RGB-D cameras," in *Robot Soccer World Cup*, 2011.
[24] J. Shi and C. Tomasi, "Good features to track," in *IEEE CVPR*, 1994.
[25] J.-Y. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Intel Corporation*, 2001.
[26] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *IEEE ICRA*, 2014.
[27] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE T-RO*, 2017.