2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Congress Center Hamburg
Sept 28 - Oct 2, 2015. Hamburg, Germany

Robustness to Lighting Variations: An RGB-D Indoor Visual Odometry Using Line Segments

Yan Lu and Dezhen Song

Abstract— **Large lighting variation challenges all visual odometry methods, even with RGB-D cameras. Here we propose a line segment-based RGB-D indoor odometry algorithm robust to lighting variation. We know line segments are abundant indoors and less sensitive to lighting change than point features. However, depth data are often noisy, corrupted or even missing for line segments which are often found on object boundaries where significant depth discontinuities occur. Our algorithm samples depth data along line segments, and uses a random sample consensus approach to identify correct depth and estimate 3D line segments. We analyze 3D line segment uncertainties and estimate camera motion by minimizing the Mahalanobis distance. In experiments we compare our method with two state-of-the-art methods including a keypoint-based approach and a dense visual odometry algorithm, under both constant and varying lighting. Our method demonstrates superior robustness to lighting change by outperforming the competing methods on 6 out of 8 long indoor sequences under varying lighting. Meanwhile our method also achieves improved accuracy even under constant lighting when tested using public data.**

I. Introduction

The emergence of RGB-D cameras (e.g. Kinect) significantly reduces costs for indoor visual odometry by providing depth measurement for pixels. If treated as a pure depth sensor, an RGB-D camera has the drawback of a limited measurement range compared to Lidar. Imagine in a long clear corridor, an RGB-D camera may only obtain a point cloud of two sidewalls, which provides no information for recovering the motion along the corridor direction. Therefore, it is important to fuse RGB information with depth data to better handle such limitations. Unfortunately, conventional visual odometry relies on image feature tracking for motion estimation, which inevitably suffers from lighting condition changes. However, different types of image features have different sensitivities to lighting variations. As illustrated in Fig. 1, only 1 scale invariant feature transform (SIFT) point match is found between a pair of images under different lighting, whereas 12 line segment matches are correctly found.

It is nontrivial to leverage line segments for RGB-D visual odometry because depth data are often noisy, corrupted or even missing for line segments on object boundaries where large depth discontinuities occur. To address the challenge, we propose a novel line segment-based RGB-D visual odometry algorithm. We devise a sampling algorithm to search for correct depth values associated with each line segment. We estimate 3D line segments from the sampled points by using a random sample consensus (RANSAC)-based filtering method. We model 3D line segment uncertainties by analyzing error propagation in the estimation process and compute camera motion by minimizing the Mahalanobis distance.

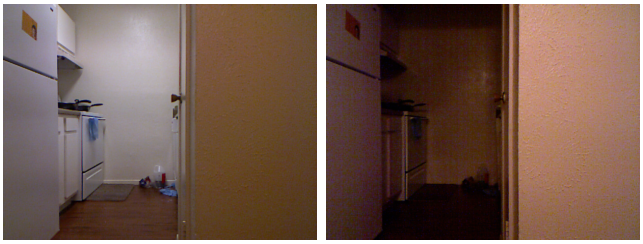


Fig. 1: Two images of the same scene under large lighting changes. SIFT detects 97 keypoints from the left image and only 58 from the right one. However, only one SIFT match is found despite the best effort from applying histogram equalization to rectify images. This renders relative motion estimation impossible. On the other hand, 112 and 94 line segments are detected from the two images using [1] and 12 matches are correctly found. (All feature transforms are set with default parameters.)

Our method has been evaluated on real-world data in experiments. We compare its performance with two state-of-the-art methods: a keypoint-based approach and a dense visual odometry algorithm, under both constant and varying lighting. Our method demonstrates superior robustness to lighting change by outperforming the competing methods on 6 out of 8 long indoor sequences under varying lighting. Meanwhile our method also achieves improved accuracy even under constant lighting when tested using public data.

II. Related Work

Our work provides an approach to visual odometry, which estimates camera motion (or poses) from a sequence of images. Visual odometry is considered as a subproblem of visual simultaneous localization and mapping (SLAM) problem in robotics.

A vast amount of visual odometry work has been developed using regular passive RGB cameras as the primary sensor, in monocular [2], stereo [3], or multi-camera configurations. To achieve high accuracy, researchers study visual odometry from different perspectives. For example, Strasdat

This work was supported in part by National Science Foundation under IIS-1318638, NRI-1426752, and NRI-1526200.

Y. Lu and D. Song are with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843, USA. Emails: {*ylu, dzsong*}*@cse.tamu.edu*

et al. [4] analyze two prevalent approaches to visual SLAM and find that bundle adjustment (e.g. [5]) produces more accurate results than sequential filtering (e.g. [6]). In the meantime, dense approaches [7] are demonstrated to permit superior accuracy than sparse feature-based algorithms, though they rely on the constant brightness assumption and need GPU acceleration for realtime purpose. Recent works also exploit inertial measurement to aid visual odometry [8].

Besides accuracy, the robustness issue is critical but lags behind in visual odometry development. Lighting variations caused by either natural or artificial lighting are well known to challenge almost every visual SLAM method [9]. Data-driven approaches have been proposed to learn lighting-invariant descriptors [10] and matching functions [11] for interest points. However, interest points are also prone to illumination variations at the detection stage. By contrast, edge and line segment detection is less sensitive to lighting changes by nature. Although edges [12], line segments [13], [14] and lines [15] have been employed for visual navigation, their accuracy is usually not as good as interest points, and their advantages in robustness are not well studied yet.

RGB-D cameras provide per-pixel depth information in addition to color images. This greatly advances the accuracy of visual odometry and leads to many dense approaches such as KinectFusion [16]–[20]. Keypoints are still the most commonly studied features. In Henry *et al.*'s RGB-D mapping system [21], keypoints are extracted from RGB images and back-projected into 3D using depth images; three point correspondences are used to find an initial pose estimation in RANSAC, and ICP is applied to further refine the result. Endres *et al.* [22] also present an RGB-D SLAM system, which uses keypoint features for camera pose estimation, achieves global consistency with pose graph optimization, and builds an octree-based volumetric map. They also test their system on the Technische Universität München (TUM) RGB-D dataset [23].

Point-based approaches are not only vulnerable to lighting variation, but also challenged in textureless environments. To overcome this shortcoming, other types of features have been recently studied. Points and planes are jointly utilized in Taguchi *et al.*'s work [24], which uses any combination of three primitives of points and planes as a minimal set for initial pose estimation in RANSAC. Planes are adopted as the primary feature in [25] for visual odometry, and points are utilized only when the number of planes is insufficient. In [26] planes are employed as the only feature for SLAM. However, the application of plane feature is limited to plane-dominant environments.

A 3D edge-based approach is recently proposed by Choi *et al.* [27], which treats the 3D edges as an intelligently-downsampled version of dense point clouds and applies the ICP algorithm for registration. Despite its potential robustness advantage, the method only evaluated the accuracy, and its dependence on ICP makes it vulnerable to initialization error and false correspondence. Moreover, this method does not actively recognize the corrupted depth values and the associated uncertainties due to discontinuities in depth values along object boundaries. Hence the method cannot achieve the best precision that line segments allow.

Our group has focused on visual navigation using passive vision systems in the past. We have studied appearance-based [28]–[30], vertical line-based [14], and heterogeneous features-based [31]–[34] visual navigation. In the process, we have learned the limitations of RGB cameras and the importance of robustness, which leads to this work.

III. Problem Definition

An RGB-D camera is usually composed of an RGB camera and a depth-measuring system. Thus, its output consists of both color images and depth images. Here we make the following assumptions.

- **a.1** The RGB camera is pre-calibrated.
- **a.2** The color and depth images are synchronized and registered with respect to each other.

Let us denote the color image by I_k and the depth image by D_k for a given time k, and define an *RGB-D frame* to be $F_k := \{I_k, D_k\}$. Assumption **a.2** implies that I_k and D_k have a spatiotemporal pixel-wise correspondence.

From an RGB-D frame F_k, we can detect a set of 3D line segments $\{\mathbf{L}_{i,k} | i = 1, 2, \cdots\}$. A 3D line segment is represented as $\mathbf{L}_{i,k} = [\mathbf{A}_{i,k}^\mathsf{T}, \mathbf{B}_{i,k}^\mathsf{T}]^\mathsf{T}$, where $\mathbf{A}_{i,k}$ and $\mathbf{B}_{i,k}$ are the two endpoints. As pair-wise motion estimation is the basic element of visual odometry, we focus our effort on the following problem.

Problem: Given two RGB-D frames F_{k_1} and F_{k_2}, compute their relative motion represented by a rotation matrix R and a translation vector t using line segment features.

IV. Line Segment-based RGB-D Odometry

The input to our approach is two RGB-D frames F_{k_1} and F_{k_2}. From each frame, we detect 3D line segments using both color and depth information and analyze the measurement uncertainties in 3D. Then we find the putative line segment correspondences between the two frames using color information, and finally estimate the relative camera motion using RANSAC. We begin with 3D line segment detection.

A. 3D Line Segment Detection and Estimation

Suppose we are given an RGB-D frame F (time subscript will be omitted in section IV-A for simplicity). We detect 3D line segments for F by considering cues from both color and depth data. Since RGB-D data contain errors, we will also analyze the error distribution for the detected 3D line segments.

1) 2D Line Segment Detection and Sampling: Under the pinhole camera model, lines remain straight when projected from 3D to images. Therefore, we start 3D line segment detection by finding line segments from the color image. Here we employ the line segment detector (LSD) [1] to extract a set of 2D line segments $\mathcal{S}_{2D} = \{\mathbf{s}_i | i = 1, 2, \cdots\}$ from I. Each line segment is represented by two endpoints $\mathbf{s}_i = [\mathbf{a}_i^\mathsf{T}, \mathbf{b}_i^\mathsf{T}]^\mathsf{T}$.

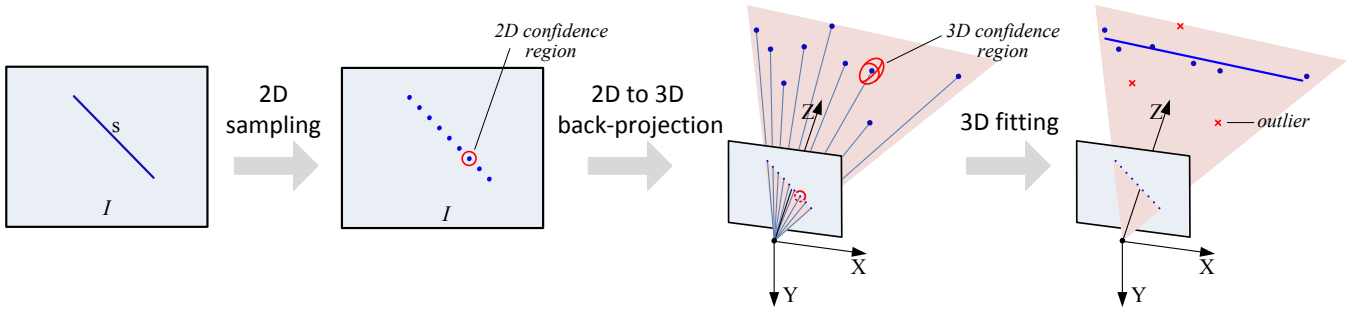


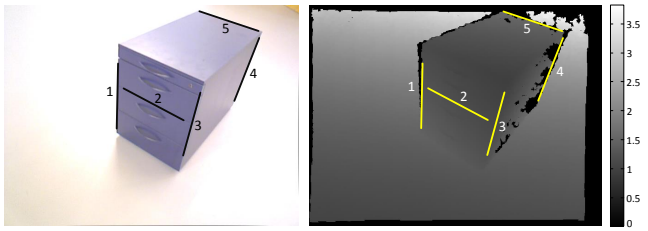
Fig. 2: Sampling-based 3D line segment estimation. From a 2D line segment $\mathbf{s}$, n_s evenly-spaced points are sampled. The sample points are back-projected to 3D using depth information. Then a 3D line segment is fitted to these 3D points using RANSAC and Mahalanobis distance-based optimization.

A naive way to obtain the 3D position of a 2D image line segment is to back-project its two endpoints to 3D using the depth map. However, this method does not work well in practice for two reasons:

- *Depth corruption*. Depth information is not always available, especially in depth-discontinuous regions. Fig. 3 shows that line segments on object boundaries (e.g., 1, 4, 5) often suffer noisy, corrupted, or totally missing depth values.
- *Linearity ambiguity*. A line segment in $\mathcal{S}_{2D}$ does not necessarily correspond to a line segment in 3D - it may also be the projection of a curved object. This ambiguity cannot be resolved by only checking the 3D positions of the two endpoints of the 2D line segment.

This suggests that we should inspect more depth values from an image line segment to avoid ambiguity and improve accuracy. However, depth measurements in D have specific error distributions determined by the depth acquisition method. This should be taken into consideration when fusing depth information with RGB data.

Thus, for a given 2D line segment $\mathbf{s}$ in I, we propose sampling n_s points evenly spaced on $\mathbf{s}$ as illustrated in Fig. 2. In all experiments, we set $n_s = \min(100, \lfloor \|\mathbf{s}\| \rfloor)$, where $\|\mathbf{s}\|$ is the length of $\mathbf{s}$ (in pixel) and $\lfloor \cdot \rfloor$ is the floor function. After removing the sample points with unavailable depth, we retain a set of points $\mathcal{G}, |\mathcal{G}| \leq n_s$. We then back-project these points to 3D, which are used to detect and estimate a 3D line segment.



(a) color image (b) depth image

Fig. 3: The depth corruption problem for line segments in RGB-D data. Five line segments are highlighted in (a) color and (b) depth images, respectively. In depth image, 0 grayscale value indicates no depth data. Observe that line segments 1, 4, and 5 suffer from the corrupted or missing depth values due to depth discontinuities.

2) Back-projecting the Sampled Points to 3D: For any $\mathbf{g}_j \in \mathcal{G}$, let $\mathbf{g}_j = [u_j, v_j]^\mathsf{T}$ and its depth value from D be d_j. The 3D position $\mathbf{G}_j$ for $\mathbf{g}_j$ in the camera coordinate system is then

$$\mathbf{G}_j := \begin{bmatrix} x_j \\ y_j \\ z_j \end{bmatrix} = \begin{bmatrix} (u_j - c_u)d_j/f \\ (v_j - c_v)d_j/f \\ d_j \end{bmatrix}, \tag{1}$$

where $[c_u, c_v]^\mathsf{T}$ and f are the principal point and focal length of the RGB camera, respectively.

As a function of $[\mathbf{g}_j^\mathsf{T}, d_j]^\mathsf{T}$, $\mathbf{G}_j$'s measurement uncertainty depends on the error distribution of $[\mathbf{g}_j^\mathsf{T}, d_j]^\mathsf{T}$. The noise distribution of $\mathbf{g}_j$ is modeled as a zero-mean Gaussian with covariance $\Sigma_g = \sigma_g^2 \mathrm{I}_2$, where I_2 is a 2×2 identity matrix. The measurement error of d_j is quite complicated and affected by many factors such as the imaging sensor, depth interpolation algorithm, and depth resolution. Taking the Kinect used in this paper for example, it is commonly agreed that the depth noise is a quadratic function of the depth [35]. That is, the standard deviation (SD) σ_{d_j} of d_j can be modeled as

$$\sigma_{d_j} = c_1 d_j^2 + c_2 d_j + c_3, \tag{2}$$

where c_1, c_2 and c_3 are constant coefficients determined empirically. We set $c_1 = 0.00273$, $c_2 = 0.00074$, and $c_3 = -0.00058$ in our experiments while the unit of d_j is meter.

Assuming the noise of $\mathbf{g}_j$ is independent of that of d_j, we have

$$\mathrm{cov}\left(\begin{bmatrix} \mathbf{g}_j \\ d_j \end{bmatrix}\right) = \begin{bmatrix} \Sigma_g & 0 \\ 0 & \sigma_{d_j}^2 \end{bmatrix}. \tag{3}$$

Under first-order approximation, we derive

$$\mathrm{cov}(\mathbf{G}_j) = J_{G_j} \mathrm{cov}\left(\begin{bmatrix} \mathbf{g}_j \\ d_j \end{bmatrix}\right) J_{G_j}^\mathsf{T}, \tag{4}$$

where

$$J_{G_j} = \frac{\partial \mathbf{G}_j}{\partial(\mathbf{g}_j, d_j)} = \begin{bmatrix} d_j/f & 0 & (u_j - c_u)/f \\ 0 & d_j/f & (v_j - c_v)/f \\ 0 & 0 & 1 \end{bmatrix}.$$

3) 3D Line Segment Detection: The 3D sample points obtained above are not necessarily from a linear object, and even if they are, they may contain outliers due to large depth errors caused by depth discontinuities.

Therefore, it is necessary to filter out outliers before estimating 3D line segments. We employ RANSAC for this purpose. Since the RANSAC procedure is well known in the field, we skip most of the details except the Mahalanobis distance between a random 3D point and a 3D line adopted here. Mahalanobis distance is widely used in computer vision because it produces the optimal estimate under Gaussian assumptions. For completeness, we briefly introduce how to compute the Mahalanobis distance [36].

Given a 3D measurement point $\mathbf{X}$ with covariance $\Sigma_{\mathbf{X}}$ and a 3D line $\mathbf{L}$ represented by two points $\mathbf{A}$ and $\mathbf{B}$, we first perform SVD on $\Sigma_{\mathbf{X}}$ and obtain $\Sigma_{\mathbf{X}} = \mathsf{U}\mathsf{D}\mathsf{U}^{\mathsf{T}}$. Then we apply an affine transform to $\mathbf{A}$ and $\mathbf{B}$ and obtain $\mathbf{A}' := \mathsf{D}^{-\frac{1}{2}}\mathsf{U}^{\mathsf{T}}(\mathbf{A}-\mathbf{X}), \mathbf{B}' := \mathsf{D}^{-\frac{1}{2}}\mathsf{U}^{\mathsf{T}}(\mathbf{B}-\mathbf{X})$. The Mahalanobis distance between $\mathbf{X}$ and $\mathbf{L}$ is

$$d_{\mathrm{M}}(\mathbf{X},\mathbf{L}) = \|\mathbf{A}'\times\mathbf{B}'\| / \|\mathbf{A}'-\mathbf{B}'\|, \tag{5}$$

where $\times$ denotes the cross product.

4) Maximum Likelihood Estimation: Suppose the size of the consensus set returned by RANSAC is n. Recall that n_s points are originally sampled from the 2D line segment $\mathbf{s}$. If n/n_s is below a threshold τ (0.6 in all experiments), it implies that we do not have sufficient depth information to retrieve the 3D position of the line segment $\mathbf{s}$. If $n/n_s \geq \tau$, we proceed to estimate the 3D line segment using a Maximum Likelihood (ML) method.

Let the consensus set be $\{\mathbf{G}_j | j = 1,\cdots,n\}$. We formulate the ML estimation problem in a way that makes the derivation of estimation uncertainty easier. Define a parameter vector $\mathbf{p} = [\hat{\mathbf{G}}_1^{\mathsf{T}}, \lambda_2, \cdots, \lambda_{n-1}, \hat{\mathbf{G}}_n^{\mathsf{T}}]^{\mathsf{T}}$, a measurement vector $\mathbf{m} = [\mathbf{G}_1^{\mathsf{T}}, \mathbf{G}_2^{\mathsf{T}}, \cdots, \mathbf{G}_n^{\mathsf{T}}]^{\mathsf{T}}$, and a measurement function

$$h(\mathbf{p}) = \begin{bmatrix} \hat{\mathbf{G}}_1 \\ \lambda_2\hat{\mathbf{G}}_1 + (1-\lambda_2)\hat{\mathbf{G}}_n \\ \vdots \\ \lambda_{n-1}\hat{\mathbf{G}}_1 + (1-\lambda_{n-1})\hat{\mathbf{G}}_n \\ \hat{\mathbf{G}}_n \end{bmatrix}.$$

In the above, $\hat{\mathbf{G}}_1$ and $\hat{\mathbf{G}}_n$ are the estimation for $\mathbf{G}_1$ and $\mathbf{G}_n$, respectively, and $\lambda_j\hat{\mathbf{G}}_1 + (1-\lambda_j)\hat{\mathbf{G}}_n$ is the estimation of $\mathbf{G}_j$ for $j = 2,\cdots,n-1$.

The parameterization of $\mathbf{p}$ ensures that the estimated points $\hat{\mathbf{G}}_j, j = 1,\cdots,n$ are collinear. The ML estimation is now an unconstrained minimization problem

$$\min_{\mathbf{p}} (\mathbf{m}-h(\mathbf{p}))^{\mathsf{T}}\Sigma_m^{-1}(\mathbf{m}-h(\mathbf{p})), \tag{6}$$

with $\Sigma_m = diag\big[\mathrm{cov}(\mathbf{G}_1),\cdots,\mathrm{cov}(\mathbf{G}_n)\big]$, where $diag[\cdots]$ returns a block diagonal matrix of the input matrices.

This problem can be solved using the Levenberg-Marquardt algorithm. Through error back-propagation [37], we obtain the covariance of estimation as

$$\mathrm{cov}(\mathbf{p}) = (J_h^{\mathsf{T}}\Sigma_m^{-1}J_h)^{-1}, \text{ with } J_h = \frac{\partial h}{\partial \mathbf{p}}. \tag{7}$$

We represent the ML estimation of the 3D line segment by $\mathbf{L} := [\mathbf{A}^{\mathsf{T}}, \mathbf{B}^{\mathsf{T}}]^{\mathsf{T}} = [\hat{\mathbf{G}}_1^{\mathsf{T}}, \hat{\mathbf{G}}_n^{\mathsf{T}}]^{\mathsf{T}}$. The covariance of $\hat{\mathbf{G}}_1$ and $\hat{\mathbf{G}}_n$ can be easily retrieved from $\mathrm{cov}(\mathbf{p})$.

B. Relative Motion Estimation

After obtaining 3D line segment observations for F_{k_1} and F_{k_2}, we need to find line segment correspondences in order to estimate relative camera motion.

Since each 3D line segment has a corresponding 2D line segment in the color image, we first perform 2D line segment matching using a method based on the line segment descriptor MSLD [38]. The resulting matches may contain false correspondences since no geometric constraints are considered.

To remove false matches and estimate the relative camera motion, RANSAC is again applied to the putative 3D line segment correspondences. In the RANSAC process, an initial relative motion is computed from two non-parallel line segment correspondences using an SVD-based algorithm proposed by [39]. Given a rotation matrix R and a translation vector t, the error metric used to determine inlier/outlier for a correspondence $\mathbf{L}_{i,k_1} \leftrightarrow \mathbf{L}_{j,k_2}$ is defined as

$$\begin{aligned} &e_{\mathrm{R,t}}(\mathbf{L}_{i,k_1}, \mathbf{L}_{j,k_2}) \\ &= d_{\mathrm{M}}(\mathrm{R}\mathbf{A}_{i,k_1}+\mathrm{t}, \mathbf{L}_{j,k_2})^2 + d_{\mathrm{M}}\left(\mathrm{R}^{\mathsf{T}}\mathbf{A}_{j,k_2}-\mathrm{R}^{\mathsf{T}}\mathrm{t}, \mathbf{L}_{i,k_1}\right)^2 \\ &+ d_{\mathrm{M}}(\mathrm{R}\mathbf{B}_{i,k_1}+\mathrm{t}, \mathbf{L}_{j,k_2})^2 + d_{\mathrm{M}}\left(\mathrm{R}^{\mathsf{T}}\mathbf{B}_{j,k_2}-\mathrm{R}^{\mathsf{T}}\mathrm{t}, \mathbf{L}_{i,k_1}\right)^2 \end{aligned} \tag{8}$$

where $d_{\mathrm{M}}(\cdot,\cdot)$ is the Mahalanobis distance defined in (5).

Let the maximum consensus set of line segment correspondences be $\mathcal{C}$. We formulate the following optimization problem to further refine the relative motion estimation,

$$(\mathrm{R}^*, \mathrm{t}^*) = \underset{\mathrm{R,t}}{\mathrm{argmin}} \sum_{\mathbf{L}_{i,k_1}\leftrightarrow\mathbf{L}_{j,k_2}\in\mathcal{C}} e_{\mathrm{R,t}}(\mathbf{L}_{i,k_1}, \mathbf{L}_{j,k_2}). \tag{9}$$

V. Experiments

We have implemented our method in C++, named line-based visual odometry (LiVO), available online [40]. We evaluate LiVO under both varying and constant lighting, and compare its performance with the following state-of-the-art algorithms.

- Kpoint: a representative keypoint based visual SLAM algorithm [22], open source software, referred to as Kpoint here.
- DVO: a recent dense visual SLAM method [18], open source software.
- Edge: the latest edge-based RGB-D method [27], referred to as Edge here. Edge is only compared on public dataset because it is not open source.

We start with evaluation under varying lighting using author-collected data [40].

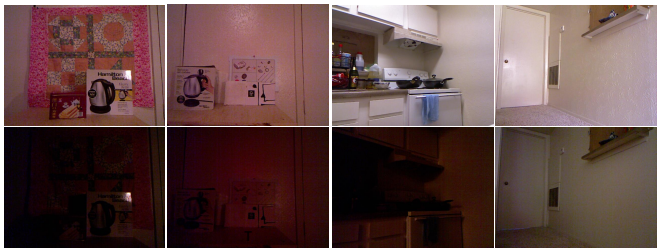


Fig. 4: Sample image pairs with lighting changes (best viewed on screen)

A. Pair-wise Motion Estimation Test Under Varying Lighting

We first investigate pair-wise motion estimation since it forms the base of visual odometry. No existing dataset allows evaluation under lighting variations. Therefore, we have collected our data using a Kinect. This dataset contains 17 pairs of RGB-D frames. Each pair is composed of two frames acquired under well- and poorly-illuminated conditions, respectively. As shown in Fig. 4, our data cover a variety of scenes, which are described by the following metrics for each pair:

- n_L, n_P: the numbers of line segments and SIFT points detected from the well-illuminated image, respectively.
- $\tau_L = n_L/(n_L+n_P)$: relative richness of line segments.
- $\gamma := \mu_p/\mu_w$, where μ_p and μ_w are the average intensity of the poorly- and well-illuminated images, respectively.
- $|\mathrm{t}|$: the true translation distance between two frames, obtained by manual measurement.

We compute the error of translation distance for evaluation as we only have ground truth for that. Table I shows the estimation results as well as the data description. Kpoint fails on 4 out of the 17 image pairs due to the failure of finding enough point correspondences. To our surprise, DVO is able to handle all image pairs. However, LiVO demonstrates clear advantages over Kpoint and DVO by producing the smallest errors on most of the pairs.

TABLE I: TRANSLATION DISTANCE ERROR (MM)

Pair	τ_L	n_P	n_L	γ	$\|\mathrm{t}\|$	Kpoint	DVO	LiVO
1	0.36	632	360	0.12	0	76.2	33.2	**15.7**
2	0.35	581	310	0.12	0	21.8	21.9	**15.9**
3	0.37	633	379	0.12	0	35.6	33.6	**16.7**
4	0.45	443	368	0.11	0	fail	26.7	**22.4**
5	0.50	392	339	0.12	0	50.4	12.6	**6.4**
6	0.60	161	240	0.17	0	fail	18.3	**16.3**
7	0.66	92	180	0.18	0	37.9	18.5	**13.2**
8	0.61	224	356	0.12	0	fail	59.3	**48.5**
9	0.60	97	143	0.75	0	**17.2**	33.8	43.5
10	0.70	103	244	0.54	0	88.1	36.2	**36.0**
11	0.50	300	293	0.15	0	fail	**41.0**	41.8
12	0.55	221	268	0.13	0	70.7	**59.1**	62.0
13	0.51	188	196	0.58	0	**37.9**	52.2	50.8
14	0.77	47	158	0.26	0	**45.2**	139.4	56.1
15	0.66	92	180	0.18	50	29.9	20.2	**10.7**
16	0.52	373	418	0.18	51	6.9	18.4	**4.2**
17	0.40	447	296	0.14	52	**5.3**	17.1	11.6

B. Visual Odometry Test Under Varying Lighting

Fig. 5: Sample images from our visual odometry dataset.

We now evaluate our method on real-world visual odometry tasks. We record RGB-D data at 30 FPS by hand-holding a Kinect and walking in typical indoor environments, including corridors, staircases and halls as illustrated in Fig. 5. The trajectory lengths, listed in Table II, range from 41 m to 86 m, which are sufficient for indoor testing. At each site, we record a pair of sequences under constant and varying lighting, respectively. Lighting variations are generated by constantly adjusting and/or swinging a hand-held dimmable LED light panel (Polaroid PL-LED350). Fig. 6 shows an example of the effect of varying lighting - while the image brightness (i.e. the average intensity of an image) varies over time even under constant lighting, the fluctuation of image brightness is significantly more intense under varying lighting. This brings great challenge for feature tracking.

We enforce the two endpoints (i.e., the starting and ending points) of each sequence to be at the same position. As a result, we define a trajectory endpoint drift (TED) to be the Euclidean distance between the two endpoints of an estimated trajectory, which serves as our evaluation metric. For fair comparison, loop closure is disabled for Kpoint and DVO since it is beyond the scope of this paper. From Table II, we can see LiVO achieves least TED on the majority of sequences, especially under varying lighting. This demonstrates the robustness of LiVO to lighting change. DVO does not perform as well as Kpoint under varying lighting because its photo-consistency assumption does not

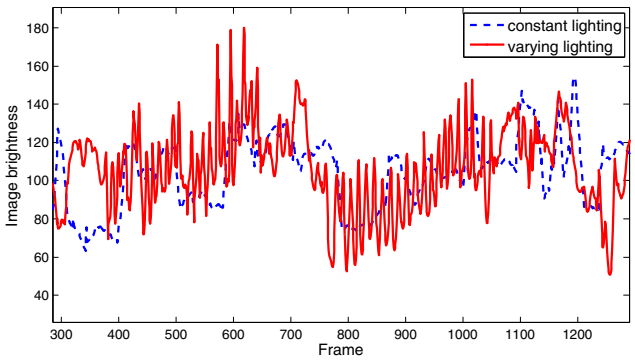


Fig. 6: Example of image brightness change over time under constant/varying lighting (data from Corridor-C). Here image brightness means the average intensity of an image.

hold.

TABLE II: TRAJECTORY ENDPOINT DRIFT (M) ON THE AUTHOR-COLLECTED VISUAL ODOMETRY DATASET

Site	Trajectory	Lighting	Kpoint	DVO	LiVO
Corridor-A	82 m	constant	**4.36**	7.10	5.47
		varying	16.68	15.41	**13.72**
Corridor-B	77 m	constant	8.25	7.56	**5.85**
		varying	12.75	12.96	**8.63**
Corridor-C	86 m	constant	6.53	6.12	**6.04**
		varying	7.30	5.93	**5.21**
Staircase-A	52 m	constant	4.04	2.26	**1.55**
		varying	4.47	3.17	**2.79**
Staircase-B	45 m	constant	5.77	**1.72**	2.81
		varying	**3.12**	3.35	3.67
Staircase-C	41 m	constant	4.51	13.87	**3.59**
		varying	8.79	16.00	**4.82**
Entry-Hall	54 m	constant	1.53	**1.31**	1.77
		varying	3.78	6.59	**3.75**
Auditorium	53 m	constant	5.78	2.39	**2.03**
		varying	**6.74**	10.66	9.76

C. Test on TUM Dataset Under Constant Lighting

We also evaluate our method under constant lighting using the TUM FR1 dataset, which is most frequently studied in the literature. The FR1 dataset consists of 9 sequences with high-precision ground truth provided, mainly covering desktop and office scenarios.

The evaluation metric used here is the relative pose error (RPE) proposed in [23]. For a given interval Δ, the RPE at time instant i is defined as

$$\mathbf{E}_i := \left(\mathbf{Q}_i^{-1}\mathbf{Q}_{i+\Delta}\right)^{-1}\left(\mathbf{P}_i^{-1}\mathbf{P}_{i+\Delta}\right), \tag{10}$$

where $\mathbf{Q}_i \in \mathrm{SE}(3)$ and $\mathbf{P}_i \in \mathrm{SE}(3)$ are the i-th ground truth and estimated poses, respectively. Specifically, we compute the root mean squared error (RMSE) of the translational RPE and that of the rotational RPE over the sequence.

In Table III, we compare our method with Kpoint and Edge, where the RPE is computed with $\Delta = 1$ frame in (10) for all methods. The errors of Kpoint are computed using their published resulting trajectories [41]. The errors of Edge are directly excerpted from [27]. For each sequence, the first and second rows represent the translational and rotational errors, respectively. In each row, we use bold font to indicate the best result among all methods. It can be seen that LiVO produces the best results on most sequences. Furthermore, we compute an average error over all sequences weighted by their frame numbers. Our method achieves the smallest average errors. Specifically, our average translational error is 37% less than that of Edge, the second smallest one.

In Table IV, we compare with DVO separately because only translational errors are reported in [18] and the unit is m/s, i.e. $\Delta = 1$ second in (10). As can be seen, our method achieves similar visual odometry accuracy as DVO does.

VI. CONCLUSION AND FUTURE WORK

To improve robustness to large lighting variations for indoor navigation, we proposed a line segment based visual odometry algorithm using an RGB-D camera. Our algorithm sampled depth data in the line segment regions, and used a random sample consensus approach to identify correct depth values and estimate 3D line segments. We analyzed 3D line segment uncertainties and estimated camera motion by minimizing the Mahalanobis distance. In experiments, our method showed significant improvements in robustness to lighting variations over state-of-the-art algorithms. Our method also achieved better accuracy under constant lighting when tested on the TUM dataset.

It is worth noting that our algorithm will fail when few line features exist in the view. Therefore, our method is not intended to fully replace any existing point-based or dense approaches. On the contrary, we envision a more accurate and robust system by properly fusing point and line segment features for RGB-D visual odometry. This is one direction we will explore in the future. We will also extend our algorithm to a full-fledged SLAM method by investigating loop closing using the combined feature types.

ACKNOWLEDGMENT

We would like to thank M. Hielsberg, J. Lee, C. Chou, H. Cheng, X. Wang, M. Treat, R. Liu and B. Chen for their inputs on this work and contributions to the Networked Robots Lab at Texas A&M University.

REFERENCES

[1] R. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 722–732, April 2010.

TABLE III: RMSE OF RPE (PER FRAME) ON FR1 SEQUENCES

Sequence	#Frame	Kpoint	Edge	LiVO
FR1 360	745	13.8 mm	**11.2** mm	**11.2** mm
		0.63 deg	**0.55** deg	0.64 deg
FR1 desk	575	11.7 mm	**8.6** mm	9.5 mm
		0.73 deg	0.70 deg	**0.69** deg
FR1 desk2	614	17.6 mm	**8.9** mm	10.6 mm
		1.07 deg	**0.7** deg	0.80 deg
FR1 floor	1214	**3.7** mm	15.7 mm	9.9 mm
		0.29 deg	0.47 deg	0.57 deg
FR1 plant	1112	20.7 mm	6.9 mm	**6.4** mm
		1.25 deg	0.49 deg	**0.44** deg
FR1 room	1332	13.7 mm	6.2 mm	**5.9** mm
		0.63 deg	0.48 deg	**0.46** deg
FR1 rpy	687	12.1 mm	7.2 mm	**6.8** mm
		0.91 deg	**0.67** deg	0.80 deg
FR1 teddy	1395	25.4 mm	36.5 mm	**11.5** mm
		1.45 deg	0.92 deg	**0.85** deg
FR1 xyz	788	5.8 mm	4.7 mm	**4.5** mm
		0.35 deg	0.41 deg	**0.29** deg
Weighted mean		14.4 mm	13.4 mm	**8.5** mm
		0.83 deg	**0.60** deg	**0.60** deg

For each sequence, the first and second rows represent the translational and rotational errors, respectively. We use bold font to indicate the best result in each row.

TABLE IV: RMSE OF TRANSLATIONAL RPE (PER SECOND) ON FR1 SEQUENCES

Sequence	DVO (m/s)	LiVO (m/s)
FR1 360	0.119	**0.095**
FR1 desk	**0.030**	0.046
FR1 desk2	**0.055**	0.085
FR1 floor	0.090	**0.032**
FR1 plant	**0.036**	**0.036**
FR1 room	0.048	**0.047**
FR1 rpy	**0.043**	0.045
FR1 teddy	0.067	**0.059**
FR1 xyz	0.024	**0.020**
Weighted mean	0.058	**0.050**

[2] J. Civera, O. G. Grasa, A. J. Davison, and J. Montiel, "1-point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry," *Journal of Field Robotics*, vol. 27, no. 5, pp. 609–631, 2010.

[3] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 652–659, June 2004.

[4] H. Strasdat, J. M. Montiel, and A. J. Davison, "Visual SLAM: Why filter?," *Image and Vision Computing*, vol. 30, no. 2, pp. 65–77, 2012.

[5] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 225–234, 2007.

[6] 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*, vol. 29, pp. 1052–1067, June 2007.

[7] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 2320–2327, 2011.

[8] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual–inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

[9] M. Meilland, A. Comport, and P. Rives, "Real-time dense visual tracking under large lighting variations," in *British Machine Vision Conference (BMVC)*, vol. 29, 2011.

[10] N. Carlevaris-Bianco and R. M. Eustice, "Learning visual feature descriptors for dynamic lighting conditions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2769–2776, 2014.

[11] A. Ranganathan, S. Matsumoto, and D. Ilstrup, "Towards illumination invariance for visual localization," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3791–3798, 2013.

[12] E. Eade and T. Drummond, "Edge landmarks in monocular SLAM," *Image and Vision Computing*, vol. 27, no. 5, pp. 588 – 596, 2009.

[13] T. Lemaire and S. Lacroix, "Monocular-vision based SLAM using line segments," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2791–2796, April 2007.

[14] J. Zhang and D. Song, "Error aware monocular visual odometry using vertical line pairs for small robots in urban areas," in *Special Track on Physically Grounded AI, AAAI*, (Atlanta, Georgia, USA), July 2010.

[15] P. Smith, I. Reid, and A. Davison, "Real-time monocular SLAM with straight lines," in *British Machine Vision Conference (BMVC)*, pp. 17–26, 2006.

[16] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 127–136, 2011.

[17] F. Steinbrucker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense RGB-D images," in *IEEE International Conference on Computer Vision (ICCV) Workshops*, pp. 719–722, 2011.

[18] 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)*, pp. 2100–2106, 2013.

[19] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. McDonald, "Robust real-time visual odometry for dense RGB-D mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5724–5731, 2013.

[20] R. F. Salas-Moreno, B. Glocker, P. H. Kelly, and A. J. Davison, "Dense planar SLAM," in *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 157–164, 2014.

[21] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using kinect-style depth cameras for dense 3D modeling of indoor environments," *International Journal of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2012.

[22] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the RGB-D SLAM system," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1691–1696, 2012.

[23] 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)*, pp. 573–580, 2012.

[24] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng, "Point-plane SLAM for hand-held 3D sensors," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5182–5189, 2013.

[25] C. Raposo, M. Lourenço, J. P. Barreto, and M. Antunes, "Plane-based odometry using an RGB-D camera," in *British Machine Vision Conference (BMVC)*, 2013.

[26] T.-k. Lee, S. Lim, S. Lee, S. An, and S.-y. Oh, "Indoor mapping using planes extracted from noisy RGB-D sensors," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1727–1733, 2012.

[27] C. Choi, A. J. Trevor, and H. I. Christensen, "RGB-D edge detection and edge-based registration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1568–1575, 2013.

[28] J. Lee, Y. Lu, and D. Song, "Planar building facade segmentation and mapping using appearance and geometric constraints," in *Intelligent Robots and Systems, 2014 IEEE/RSJ International Conference on*, IEEE, 2014.

[29] W. Li and D. Song, "Toward featureless visual navigation: Simultaneous localization and planar surface extraction using motion vectors in video streams," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, IEEE, 2014.

[30] W. Li and D. Song, "Featureless motion vector-based simultaneous localization, planar surface extraction, and moving obstacle tracking," in *The Eleventh International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2014.

[31] H. Li, D. Song, Y. Lu, and J. Liu, "A two-view based multilayer feature graph for robot navigation," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3580–3587, May 2012.

[32] Y. Lu, D. Song, Y. Xu, A. G. A. Perera, and S. Oh, "Automatic building exterior mapping using multilayer feature graphs," in *IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 162–167, 2013.

[33] Y. Lu, D. Song, and J. Yi, "High level landmark-based visual navigation using unsupervised geometric constraints in local bundle adjustment," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1540–1545, 2014.

[34] Y. Lu and D. Song, "Visual navigation using heterogeneous landmarks and unsupervised geometric constraints," *IEEE Transactions on Robotics*, vol. 31, pp. 736–749, June 2015.

[35] J. Smisek, M. Jancosek, and T. Pajdla, "3D with Kinect," in *IEEE International Conference on Computer Vision (ICCV) Workshops*, pp. 1154–1160, 2011.

[36] Z. Lu, S. Baek, and S. Lee, "Robust 3D line extraction from stereo point clouds," in *Robotics, Automation and Mechatronics, 2008 IEEE Conference on*, pp. 1–5, IEEE, 2008.

[37] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge Univ Pr, 2003.

[38] Z. Wang, F. Wu, and Z. Hu, "MSLD: A robust descriptor for line matching," *Pattern Recognition*, vol. 42, no. 5, pp. 941–953, 2009.

[39] Z. Zhang and O. D. Faugeras, "Determining motion from 3D line segment matches: A comparative study," *Image and Vision Computing*, vol. 9, no. 1, pp. 10–19, 1991.

[40] Y. Lu and D. Song, "RGB-D odometry using line segments." `http://telerobot.cs.tamu.edu/MFG/rgbd/livo`, 2014.

[41] "RGBDSLAM trajectories." `https://svncvpr.in.tum.de/cvpr-ros-pkg/trunk/rgbd_benchmark/rgbd_benchmark_tools/data/rgbdslam/`.