Robust RGB-D Odometry Using Point and Line Features

Yan Lu Dezhen Song
Texas A&M University, College Station, TX, USA
{ylu, dzsong}@cs.tamu.edu *

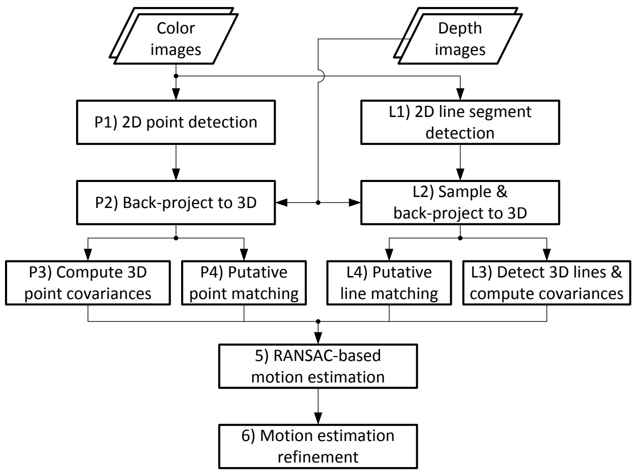
Abstract

Lighting variation and uneven feature distribution are main challenges for indoor RGB-D visual odometry where color information is often combined with depth information. To meet the challenges, we fuse point and line features to form a robust odometry algorithm. Line features are abundant indoors and less sensitive to lighting change than points. We extract 3D points and lines from RGB-D data, analyze their measurement uncertainties, and compute camera motion using maximum likelihood estimation. We prove that fusing points and lines produces smaller motion estimate uncertainty than using either feature type alone. In experiments we compare our method with state-of-the-art methods including a keypoint-based approach and a dense visual odometry algorithm. Our method outperforms the counterparts under both constant and varying lighting conditions. Specifically, our method achieves an average translational error that is 34.9% smaller than the counterparts, when tested using public datasets.

1. Introduction

For GPS-denied indoor environments, visual odometry is an attractive, low-cost alternative to laser-based robot localization approaches. The recent emergence of RGB-D cameras (*e.g.* Kinect) significantly enhances visual odometry performance by providing pixel-wise depth measurements. Besides the relative short range and the limited accuracy of existing RGB-D technologies, the main challenges come from *large lighting condition variations* and *uneven feature distributions*. The former directly hinders direct approaches (*e.g.* the dense method in [12]) which are based on the photo-consistency assumption. The latter often corrupts feature tracking quality in feature-based approaches.

Building on feature-based approaches and with the challenges in mind, we propose a robust RGB-D odometry method by fusing point and line features (see Figure 1). Line features are abundant indoors and less sensitive to lighting variation than points. On the other hand, points provide less position ambiguity than that of lines if sufficient observations are available. Effectively combining those desirable properties would increase both accuracy and robustness for an odometry method. We extract 3D points and lines from RGB-D data and analyze their measurement uncertainties. We provide a framework that seamlessly fuses points and lines by adopting a RANSAC-based motion estimation, followed by a maximum likelihood estimation (MLE)-based motion refinement. Under Gaussian noise assumption, we prove that fusing points and lines results in smaller uncertainty in motion estimation than using either feature type alone.


Color images
Depth images
P1) 2D point detection
L1) 2D line segment detection
P2) Back-project to 3D
L2) Sample & back-project to 3D
P3) Compute 3D point covariances
P4) Putative point matching
L4) Putative line matching
L3) Detect 3D lines & compute covariances
5) RANSAC-based motion estimation
6) Motion estimation refinement
Figure 1: System diagram.

Our method has been evaluated on real-world data in experiments. We compare its performance with state-of-the-art methods including a keypoint-based approach and a dense visual odometry algorithm. Our method outperforms the counterparts under both constant and varying lighting conditions. Specifically, our method achieves an average translational error that is 34.9% smaller than the counterparts, when tested using public datasets.

2. Related work

Our work belongs to visual odometry, which estimates camera trajectories (or poses) from a sequence of images. Visual odometry is considered as a subproblem of visual

*This work was supported in part by National Science Foundation under IIS-1318638, NRI-1426752, and NRI-1526200, and in part by TxDot 0-6869.

simultaneous localization and mapping (SLAM).

Many visual odometry works have been developed using regular passive RGB cameras as the primary sensor, in monocular [18, 20], stereo [24], or multi-camera settings. To improve accuracy, researchers study visual odometry from different perspectives. For example, Strasdat *et al.* [31] analyze two prevalent approaches to visual SLAM and find that bundle adjustment (e.g. [13]) produces more accurate results than sequential filtering (e.g. [6, 19]). Due to depth ambiguity, monocular visual odometry inevitably suffers from scale drift, which can be easily avoided by using an RGB-D camera [30]. Besides accuracy, robustness is another critical issue but lags behind in visual odometry development. Lighting variation and uneven feature distribution are two main challenges for robustness.

Lighting variations caused by either natural or artificial lighting challenges both direct visual odometry and feature-based methods [21]. Although direct approaches can achieve superior accuracy by doing pixel-wise registration [12, 23, 37], their fundamental assumption on photo-consistency makes them sensitive to lighting condition changes. In the feature-based category, data-driven approaches are proposed to learn lighting-invariant descriptors [4] and matching functions [25] for robust matching of feature points. However, point features are also prone to illumination variations at the detection stage. On the other hand, the detection of edge and line features is less sensitive to lighting changes by nature. Edges [7], line segments [14] and lines [29] have been applied to visual odometry/SLAM, though their accuracy is usually not comparable with that of point features. In addition to regular approaches, RGB-D odometry can also utilize point could registration methods, which originate from Lidar-based SLAM. This kind of method [22] is invariant to lighting changes, but the problem is that it easily fails in degenerated cases, e.g. when a plane dominates the scene.

Meanwhile, uneven feature distribution hinders all feature-based visual odometry algorithms. In RGB-D odometry, points are the most popular type of visual feature. For example, in Henry *et al.*'s RGB-D mapping system [10], keypoints are extracted from RGB images and back-projected into 3D using depth data. Endres *et al.* [8] present an open-source RGB-D SLAM system based on point features. These approaches can be drastically affected if the distribution of point features is largely uneven, e.g. when textureless surfaces dominate the scene. To overcome this shortcoming, other types of features are investigated in RGB-D odometry. Points and planes are jointly utilized in Taguchi *et al.*'s work [33], 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 [26] for visual odometry, and points are utilized only when the number of planes is insufficient. However, the applications of these methods are limited to plane-dominant environments. A 3D edge-based approach is proposed by Choi *et al.* [5], which treats edges as down-sampled point clouds and uses ICP. In [17] we detect 3D lines and analyze their uncertainties for RGB-D odometry. Here we further fuse line features with point features to improve the robustness of RGB-D odometry. We analyze the uncertainties of motion estimation to show the benefit of feature fusion.

3. Problem description and system overview

We assume the RGB-D camera to

a.1 be pre-calibrated, with lens distortion removed, and

a.2 have its depth image pixel-wisely synchronized with the corresponding color image.

Define an RGB-D frame at time k to be $F_k := \{I_k, D_k\}$, where I_k and D_k denote the color and depth images, respectively. The local coordinate system of F_k is the same as the RGB camera coordinate system (right-handed, Z-axis passing the camera center pointing forward, and X-axis pointing rightward). We define our problem as follows.

Problem 1 (Visual odometry). *Given an RGB-D sequence* $\{F_k\}, k \geq 1$, *estimate the camera pose of each frame with respect to a world coordinate system.*

To solve Problem 1, we estimate camera motion using adjacent RGB-D pairs, namely, F and F'. We compute a 3D rigid transformation between F and F'.

Our system (see Figure 1) mainly consists of feature detection and motion estimation. In the feature detection stage, for each RGB-D frame we detect point and line features from the color image, back-project them to 3D, and analyze their uncertainties in parallel. In the motion estimation stage, we find feature matching between two RGB-D frames and estimate the relative motion using points and lines in a joint manner.

4. Feature detection & uncertainty analysis

Errors inevitably enter the system at the feature detection stage. Eventually, the errors propagate to motion estimation results. For a deep understanding of the system accuracy, we begin with uncertainty analysis for each feature type.

4.1. Point detection & uncertainty analysis

Detection. Given an RGB-D frame F, we first detect a set of 2D points from color image I using interest point detection algorithms such as SURF [3] (see Box P1, Figure 1). Then we find the depth values, if available, for these 2D points from D. Supposing a 2D point $\mathbf{p} = [u, v]^\mathsf{T}$ in I has depth d, its 3D position w.r.t. F is computed as follows

(see Box P2, Figure 1)

$$\mathbf{P} := \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} (u - c_u)d/f_c \\ (v - c_v)d/f_c \\ d \end{bmatrix}, \tag{1}$$

where $[c_u, c_v]^T$ and f_c are the principal point and focal length of the RGB camera, respectively.

Uncertainty. As a function of $\left[\mathbf{p}^\top, d\right]^\top$, $\mathbf{P}$ has a measurement uncertainty depending on the error distribution of $\left[\mathbf{p}^\top, d\right]^\top$. The noise distribution of $\mathbf{p}$ is modeled as a zero-mean Gaussian with covariance $\sigma_p^2\, \mathbf{I}_2$, where $\mathbf{I}_2$ is a 2×2 identity matrix. The measurement error of d is determined 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 itself [28]. Specifically, the standard deviation (SD) σ_d of d is modeled as

$$\sigma_d = c_1 d^2 + c_2 d + c_3, \tag{2}$$

where c_1, c_2 and c_3 are constant coefficients. We set $c_1 = 2.73\times10^{-3}$, $c_2 = 7.4\times10^{-4}$, and $c_3 = -5.8\times10^{-4}$ in our experiments and the unit of d is meter [28].

Assuming the measurement noise of $\mathbf{p}$ is independent of that of d, we have

$$\operatorname{cov}\left(\begin{bmatrix} \mathbf{p} \\ d \end{bmatrix}\right) = \begin{bmatrix} \sigma_p^2 \mathbf{I}_2 & \mathbf{0}_{2\times1} \\ \mathbf{0}_{1\times2} & \sigma_d^2 \end{bmatrix}, \tag{3}$$

where $\operatorname{cov}(\cdot)$ indicates the covariance matrix of a random variable, and $\mathbf{0}_{m\times n}$ means a zero matrix of size $m \times n$. Under first-order approximation, we have (see Box P3, Figure 1)

$$\operatorname{cov}(\mathbf{P}) = J_P \operatorname{cov}\left(\begin{bmatrix} \mathbf{p} \\ d \end{bmatrix}\right) J_P^\top, \tag{4}$$

where $J_P = \frac{\partial \mathbf{P}}{\partial(\mathbf{p},d)} = \begin{bmatrix} d/f_c & 0 & (u - c_u)/f_c \\ 0 & d/f_c & (v - c_v)/f_c \\ 0 & 0 & 1 \end{bmatrix}$.

4.2. Line detection & uncertainty analysis

In this section, we introduce a 3D line detection method by considering cues from both color and depth data. To handle RGB-D data noise, we also present how to optimally estimate the detected 3D lines and analyze the uncertainties of the estimates. Our method starts from 2D line detection.

4.2.1 Line detection in 2D and 3D

Under the pinhole camera model, lines remain straight when projected from 3D to images. Therefore, to detect 3D lines we first detect their projections in the color image I (see Box L1, Figure 1). As long as a 3D line is visible in I, it appears as a 2D line segment. Here we employ the line segment detector LSD [35] 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 $\mathbf{s}_i = \left[\mathbf{a}_i^\top, \mathbf{b}_i^\top\right]^\top$.

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: 1) *Depth corruption*: depth information is not always available, especially on object boundaries when the depth is discontinuous, and 2) *Perspective ambiguity* because a line segment in $\mathcal{S}_{2D}$ does not necessarily correspond to a line segment in 3D as a result of the perspective projection. 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 information of a 2D line segment to avoid the aforementioned issues. As a line segment consists of infinite number of points, we propose a sampling based method for 3D line detection.

Sampling. Given a 2D line segment $\mathbf{s}$, we sample n_s points evenly spaced on $\mathbf{s}$ as illustrated in Figure 2. In all experiments, we set $n_s = \min(100, \lfloor \|\mathbf{s}\| \rfloor)$, where $\|\mathbf{s}\|$ denotes the length of $\mathbf{s}$ (in pixels) and $\lfloor \cdot \rfloor$ is the floor function. We discard the sample points whose depths are unavailable, back-project the remaining points to 3D (see Box L2, Figure 1) using (1), and compute their 3D uncertainties using (4).

The 3D sample points obtained above are not necessarily from a 3D line, and even if they are, they may contain outliers due to large depth errors. As illustrated in Figure 2, we apply RANSAC to detect the existence of 3D line segments and filter out outliers (see Box L3, Figure 1). For brevity, we skip every detail of RANSAC but the error metric used for identifying inlier/outlier. Given a 3D line and a 3D point observation (subject to measurement noise), we utilize the

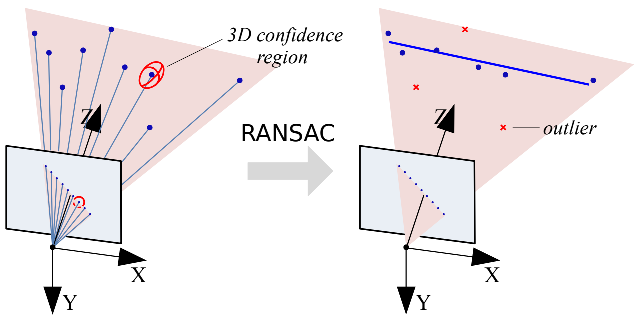


Figure 2: Sampling-based 3D line segment estimation. From a 2D line segment, n_s evenly-spaced points are sampled. The sample points are back-projected to 3D using depth information. Then a 3D line segment is fitted to these 3D points using RANSAC and Mahalanobis distance-based optimization.

Mahalanobis distance between them to evaluate whether the point is an observation of a point on the line. Mahalanobis distance is widely used in computer vision because it produces the optimal estimate under Gaussian assumptions [9]. For completeness, we briefly describe how to compute the Mahalanobis distance.

Mahalanobis distance. Let $\mathbf{P}$ be a 3D point measurement with covariance Σ_p and $\mathbf{L}$ be an infinite 3D line. The Mahalanobis distance from $\mathbf{P}$ to $\mathbf{L}$ is defined as

$$d_{\text{MAH}}(\mathbf{P},\mathbf{L}) = \min_{\mathbf{Q}\in\mathbf{L}} \sqrt{(\mathbf{P}-\mathbf{Q})^\mathsf{T}\Sigma_p^{-1}(\mathbf{P}-\mathbf{Q})}, \tag{5}$$

where $\mathbf{Q} \in \mathbf{L}$ indicates an arbitrary point lying on line $\mathbf{L}$. To derive $d_{\text{MAH}}(\mathbf{P},\mathbf{L})$, let $\mathbf{A}$ and $\mathbf{B}$ be two reference points on $\mathbf{L}$. Write $\mathbf{Q} = \mathbf{A}+\lambda(\mathbf{B}-\mathbf{A}), \lambda \in \mathbb{R}$. The minimization problem in (5) is then equivalent to minimizing the following univariate quadratic function.

$$\min_{\lambda} \begin{array}{c} \lambda^2(\mathbf{B}-\mathbf{A})^\mathsf{T}\Sigma_p^{-1}(\mathbf{B}-\mathbf{A}) + 2\lambda(\mathbf{B}-\mathbf{A})^\mathsf{T}\Sigma_p^{-1}(\mathbf{A}-\mathbf{P}) \\ +(\mathbf{A}-\mathbf{P})^\mathsf{T}\Sigma_p^{-1}(\mathbf{A}-\mathbf{P}) \end{array}$$

The optimal value of λ yields the optimal $\mathbf{Q}$ for (5) $\mathbf{Q}^* = \mathbf{A} - \frac{(\mathbf{B}-\mathbf{A})^\mathsf{T}\Sigma_p^{-1}(\mathbf{A}-\mathbf{P})}{(\mathbf{B}-\mathbf{A})^\mathsf{T}\Sigma_p^{-1}(\mathbf{B}-\mathbf{A})}(\mathbf{B}-\mathbf{A})$. Let

$$\boldsymbol{\delta}(\mathbf{P},\mathbf{L}) = \mathbf{P} - \mathbf{Q}^*, \tag{6}$$

and then we have,

$$d_{\text{MAH}}(\mathbf{P},\mathbf{L}) = \sqrt{\boldsymbol{\delta}(\mathbf{P},\mathbf{L})^\mathsf{T}\Sigma_p^{-1}\boldsymbol{\delta}(\mathbf{P},\mathbf{L})}. \tag{7}$$

4.2.2 Line uncertainty under MLE

Suppose the size of the largest consensus set returned by the aforementioned RANSAC process is n_{con}. Recall that n_s points are originally sampled from the 2D line segment $\mathbf{s}$. If n_{con}/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 $n_{\text{con}}/n_s \geq \tau$, we proceed to apply MLE to obtain the 3D line segment.

Let the largest consensus set be $\{\mathbf{G}_i | i = 1, \cdots, n_{\text{con}}\}$ with $\mathbf{G}_1$ and $\mathbf{G}_{\text{con}}$ being the two extremities. We parametrize the 3D line segment $\mathbf{L} = \left[\mathbf{A}^\mathsf{T}, \mathbf{B}^\mathsf{T}\right]^\mathsf{T}$ to be estimated by two 3D points associated with $\mathbf{G}_1$ and $\mathbf{G}_{\text{con}}$. Define a measurement error function

$$w(\mathbf{L}) = \begin{bmatrix} \mathbf{G}_1 - \mathbf{A} \\ \boldsymbol{\delta}(\mathbf{G}_2, \mathbf{L}) \\ \vdots \\ \boldsymbol{\delta}(\mathbf{G}_{n_{\text{con}-1}}, \mathbf{L}) \\ \mathbf{G}_{n_{\text{con}}} - \mathbf{B} \end{bmatrix}, \tag{8}$$

where $\boldsymbol{\delta}(\cdot,\cdot)$ is defined in (6). The MLE of $\mathbf{L}$ is obtained as follows,

$$\mathbf{L}^* = \min_{\mathbf{L}} w(\mathbf{L})^\mathsf{T}\Sigma_w^{-1}w(\mathbf{L}), \tag{9}$$

where $\Sigma_w = diag\big(\text{cov}(\mathbf{G}_1), \cdots, \text{cov}\,(\mathbf{G}_{n_{\text{con}}})\big)$ is a block-wise diagonal matrix. This problem is then solved using the Levenberg-Marquardt (LM) algorithm. From back-propagation of covariance [9], we obtain the covariance of the MLE (see Box L3, Figure 1)

$$\text{cov}(\mathbf{L}^*) = \left(J_w^\mathsf{T}\Sigma_w^{-1}J_w\right)^{-1}, \tag{10}$$

where $J_w = \frac{\partial w}{\partial \mathbf{L}}\big|_{\mathbf{L}=\mathbf{L}^*}$.

5. Motion estimation & uncertainty analysis

With point and line features detected and the understanding of their error covariance, we are ready to perform overall camera motion estimation for the adjacent frame pair and analyze the uncertainty of the estimation.

5.1. Putative feature matching and RANSAC-based motion estimation

Once 3D points and lines are detected from F and F', we need to find feature correspondences between frames. As both 3D points and lines have associated 2D points and lines, we do feature matching using 2D features (see Boxes P4 and L4, Figure 1). We compute the SURF descriptors [3] for points and the MSLD descriptors [36] for lines, respectively. We adopt nearest-neighbour method in descriptor space for the putative matching.

As putative matching inevitably contains false matching, we use RANSAC to filter out outliers and estimate the 3D rigid transformation (see Box 5, Figure 1). Up to this point, points and lines have been processed in parallel. However, in each iteration of RANSAC, a minimal set of data is randomly sampled for a motion estimate; a seamless fusion of points and lines should allow mixed features in a minimal set. For this purpose, we consider the four possible types of minimal sets as follows,

- **3 point** matches. This can be trivially solved using methods like [2, 34].
- **3 line** matches. In fact the 3D rigid transformation can be recovered by only 2 line matches using an SVD-based method [38]. Considering that this method is very sensitive to noise, we sample 3 line matches.
- **1 point + 2 line** matches. In each frame, we orthogonally project the point onto the 2 lines, respectively, which converts this case to a case of "3 point" matches.
- **2 point + 1 line** matches. We choose one point and orthogonally project it onto the line in each frame, converting this case to a case of "3 point" matches.

5.2. MLE of motion and uncertainty analysis

The RANSAC process results in a largest consensus set of feature matches consisting of both point matches and line matches. We refine the motion estimate with the whole consensus set of point and line matches using MLE (see Box 6, Figure 1). We prove that the MLE of 3D rigid transformation obtained using both types of feature correspondences has smaller uncertainty than that obtained using either type of feature correspondence alone. We start by deriving the uncertainties for the MLE of motion obtained using points and lines separately before fusing them.

5.2.1 Point-based motion estimation

Let the set of 3D point correspondences between F and F' be $\{\mathbf{P}_i \leftrightarrow \mathbf{P}'_i, i = 1, \cdots, n\}$, where $n \geq 3$. Denote the covariance of $\mathbf{P}_i$ and $\mathbf{P}'_i$ by Σ_i and Σ'_i, respectively. The rigid body transformation $\mathbf{T}$ is parametrized by a six-vector $\boldsymbol{\xi}$. $\mathbf{T}$ can also represented by rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$,

$$\mathbf{T}(\mathbf{X}) := \mathbf{R}\mathbf{X} + \mathbf{t}, \tag{11}$$

where $\mathbf{X}$ is a 3D point.

To achieve an MLE of motion, the underlying 3D point landmarks must be estimated simultaneously. Let $\mathbf{X}_i$ be the 3D point with respect to F that underlies the measurements $\mathbf{P}_i$ and $\mathbf{P}'_i$. The parameter vector to be estimated is thus defined as $\mathbf{p} = \left[\boldsymbol{\xi}^\mathsf{T}, \mathbf{X}_1^\mathsf{T}, \cdots, \mathbf{X}_n^\mathsf{T}\right]^\mathsf{T}$.

Define a measurement error function

$$h(\mathbf{p}) = \begin{bmatrix} \mathbf{h_p} \\ \mathbf{h'_p} \end{bmatrix}, \tag{12}$$

where $\mathbf{h_p} = \begin{bmatrix} \mathbf{X}_1 - \mathbf{P}_1 \\ \vdots \\ \mathbf{X}_n - \mathbf{P}_n \end{bmatrix}$ and $\mathbf{h'_p} = \begin{bmatrix} \mathbf{T}(\mathbf{X}_1) - \mathbf{P}'_1 \\ \vdots \\ \mathbf{T}(\mathbf{X}_n) - \mathbf{P}'_n \end{bmatrix}$.

The MLE of motion solves the following problem

$$\min_{\mathbf{p}} h(\mathbf{p})^\mathsf{T} \Sigma_h^{-1} h(\mathbf{p}), \tag{13}$$

where $\Sigma_h = diag\left(\Sigma_1, \cdots, \Sigma_n, \Sigma'_1 \cdots, \Sigma'_n\right)$. Lemma 1 provides the estimation uncertainty of motion.

Lemma 1. *Under Gaussian noise assumption, the point feature-based MLE of rigid transformation $\boldsymbol{\xi}$ obtained by (13) has covariance*

$$C_P = \left(H_h^A - H_h^B {H_h^D}^{-1} {H_h^B}^\mathsf{T}\right)^{-1}, \tag{14}$$

where

$$H_h^A = \sum_{i=1}^n \left(\frac{\partial \mathbf{T}(\mathbf{X}_i)}{\partial \boldsymbol{\xi}}\right)^\mathsf{T} \Sigma_i'^{-1} \frac{\partial \mathbf{T}(\mathbf{X}_i)}{\partial \boldsymbol{\xi}},$$

$$H_h^B = \begin{bmatrix} \left(\frac{\partial \mathbf{T}(\mathbf{X}_1)}{\partial \mathbf{X}_1}\right)^\mathsf{T} \Sigma_1'^{-1} \frac{\partial \mathbf{T}(\mathbf{X}_1)}{\partial \boldsymbol{\xi}} \\ \vdots \\ \left(\frac{\partial \mathbf{T}(\mathbf{X}_n)}{\partial \mathbf{X}_n}\right)^\mathsf{T} \Sigma_n'^{-1} \frac{\partial \mathbf{T}(\mathbf{X}_n)}{\partial \boldsymbol{\xi}} \end{bmatrix}^\mathsf{T}, \quad \textit{and}$$

$$H_h^D = diag\Bigg(\Sigma_1^{-1} + \left(\frac{\partial \mathbf{T}(\mathbf{X}_1)}{\partial \mathbf{X}_1}\right)^\mathsf{T} \Sigma_1'^{-1} \frac{\partial \mathbf{T}(\mathbf{X}_1)}{\partial \mathbf{X}_1}, \cdots, \Sigma_n^{-1} + \left(\frac{\partial \mathbf{T}(\mathbf{X}_n)}{\partial \mathbf{X}_n}\right)^\mathsf{T} \Sigma_n'^{-1} \frac{\partial \mathbf{T}(\mathbf{X}_n)}{\partial \mathbf{X}_n}\Bigg).$$

Proof. By back-propagation of covariance, the MLE of $\mathbf{p}$ has covariance [9]

$$\text{cov}(\mathbf{p}) = \left(J_h^\mathsf{T} \Sigma_h^{-1} J_h\right)^{-1}, \tag{15}$$

with

$$J_h = \frac{\partial h}{\partial \mathbf{p}} = \left[\begin{array}{c:c} J_h^A & J_h^B \\ \hdashline J_h^C & J_h^D \end{array}\right] = \left[\begin{array}{c:ccc} \mathbf{0} & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_3 \\ \hdashline \frac{\partial \mathbf{T}(\mathbf{X}_1)}{\partial \boldsymbol{\xi}} & \frac{\partial \mathbf{T}(\mathbf{X}_1)}{\partial \mathbf{X}_1} & \mathbf{0} & \mathbf{0} \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \frac{\partial \mathbf{T}(\mathbf{X}_n)}{\partial \boldsymbol{\xi}} & \mathbf{0} & \mathbf{0} & \frac{\partial \mathbf{T}(\mathbf{X}_n)}{\partial \mathbf{X}_n} \end{array}\right]_{6n \times 3n+6}$$

where $\mathbf{I}_3$ is a 3×3 identity matrix, and $\mathbf{0}$ indicates a zero matrix of a context-determined size hereafter.

With $\Sigma_h^{-1} = diag\left(\Sigma_1^{-1}, \cdots, \Sigma_n^{-1}, \Sigma_1'^{-1} \cdots, \Sigma_n'^{-1}\right)$, we derive $J_h^\mathsf{T} \Sigma_h^{-1} J_h = \left[\begin{array}{c:c} H_h^A & H_h^B \\ \hdashline {H_h^B}^\mathsf{T} & H_h^D \end{array}\right]$. Performing blockwise matrix inversion on $J_h^\mathsf{T} \Sigma_h^{-1} J_h$ yields (14). □

5.2.2 Line-based motion estimation

Let the set of 3D line correspondences between F and F' be $\{\mathbf{L}_i \leftrightarrow \mathbf{L}'_i, i = 1, \cdots, m\}$, where $m \geq 3$. Recall that $\mathbf{L}_i = \left[\mathbf{A}_i^\mathsf{T}, \mathbf{B}_i^\mathsf{T}\right]^\mathsf{T}$, $\mathbf{L}'_i = \left[\mathbf{A}_i'^\mathsf{T}, \mathbf{B}_i'^\mathsf{T}\right]^\mathsf{T}$. For simplicity, we denote $\Lambda_i = \text{cov}(\mathbf{L})$, $\Lambda'_i = \text{cov}(\mathbf{L}'_i)$ (see (10)).

For MLE of motion, the underlying 3D line landmarks must be estimated simultaneously. Let $\mathbf{Y}_i = \left[\boldsymbol{\alpha}_i^\mathsf{T}, \boldsymbol{\beta}_i^\mathsf{T}\right]^\mathsf{T}$ be the 3D line with respect to F that underlies the measurements $\mathbf{L}_i$ and $\mathbf{L}'_i$. The parameter vector to be estimated is thus defined as $\mathbf{q} = [\boldsymbol{\xi}^\mathsf{T}, \mathbf{Y}_1^\mathsf{T}, \cdots, \mathbf{Y}_m^\mathsf{T}]^\mathsf{T}$.

Recall $\boldsymbol{\delta}(\mathbf{A}_i, \mathbf{Y})$ is a 3-vector function. Define

$$\boldsymbol{\eta}(\mathbf{L}_i, \mathbf{Y}_i) = \left[\boldsymbol{\delta}(\mathbf{A}_i, \mathbf{Y}_i)^\mathsf{T}, \boldsymbol{\delta}(\mathbf{B}_i, \mathbf{Y}_i)^\mathsf{T}\right]^\mathsf{T}. \tag{16}$$

Define a measurement error function

$$g(\mathbf{q}) = \begin{bmatrix} \mathbf{g_l} \\ \mathbf{g'_l} \end{bmatrix}, \tag{17}$$

where $\mathbf{g_l} = \begin{bmatrix} \boldsymbol{\eta}(\mathbf{L}_1, \mathbf{Y}_1) \\ \vdots \\ \boldsymbol{\eta}(\mathbf{L}_m, \mathbf{Y}_m) \end{bmatrix}$, $\mathbf{g'_l} = \begin{bmatrix} \boldsymbol{\eta}(\mathbf{L}'_1, \mathbf{T}(\mathbf{Y}_1)) \\ \vdots \\ \boldsymbol{\eta}(\mathbf{L}'_m, \mathbf{T}(\mathbf{Y}_m)) \end{bmatrix}$, and $\mathbf{T}(\mathbf{Y}_i) := \left[\mathbf{T}(\boldsymbol{\alpha}_i)^\mathsf{T}, \mathbf{T}(\boldsymbol{\beta}_i)^\mathsf{T}\right]^\mathsf{T}$. The MLE solves the following problem

$$\min_{\mathbf{q}} g(\mathbf{q})^\mathsf{T} \Sigma_g^{-1} g(\mathbf{q}), \tag{18}$$

where $\Sigma_g = diag\left(\Lambda_1, \cdots, \Lambda_m, \Lambda'_1, \cdots, \Lambda'_m\right).$

Lemma 2. *Under Gaussian noise assumption, the line feature-based MLE of rigid transformation $\boldsymbol{\xi}$ obtained by (18) has covariance*

$$C_L = \left(H_g^A - H_g^B {H_g^D}^{-1} {H_g^B}^\mathsf{T}\right)^{-1}, \tag{19}$$

where

$$H_g^A = \sum_i \left(\frac{\partial \boldsymbol{\eta}(\mathbf{L}'_i, \mathbf{T}(\mathbf{Y}_i))}{\partial \boldsymbol{\xi}}\right)^\mathsf{T} \Lambda_i'^{-1} \frac{\partial \boldsymbol{\eta}(\mathbf{L}'_i, \mathbf{T}(\mathbf{Y}_i))}{\partial \boldsymbol{\xi}},$$

$$H_g^B = \begin{bmatrix} \left(\frac{\partial \boldsymbol{\eta}(\mathbf{L}'_1, \mathbf{T}(\mathbf{Y}_1))}{\partial \mathbf{Y}_1}\right)^\mathsf{T} \Lambda_1'^{-1} \frac{\partial \boldsymbol{\eta}(\mathbf{L}'_1, \mathbf{T}(\mathbf{Y}_1))}{\partial \boldsymbol{\xi}} \\ \vdots \\ \left(\frac{\partial \boldsymbol{\eta}(\mathbf{L}'_m, \mathbf{T}(\mathbf{Y}_m))}{\partial \mathbf{Y}_m}\right)^\mathsf{T} \Lambda_m'^{-1} \frac{\partial \boldsymbol{\eta}(\mathbf{L}'_m, \mathbf{T}(\mathbf{Y}_m))}{\partial \boldsymbol{\xi}} \end{bmatrix}^\mathsf{T},$$

$H_g^D = diag(\mathbf{C}_1, \cdots, \mathbf{C}_m)$, *and for* $i = 1, \cdots, m$

$$\mathbf{C}_i = \left(\frac{\partial \boldsymbol{\eta}(\mathbf{L}_i, \mathbf{Y}_i)}{\partial \mathbf{Y}_i}\right)^\mathsf{T} \Lambda_i^{-1} \frac{\partial \boldsymbol{\eta}(\mathbf{L}_i, \mathbf{Y}_i)}{\partial \mathbf{Y}_i} + \left(\frac{\partial \boldsymbol{\eta}(\mathbf{L}'_i, \mathbf{T}(\mathbf{Y}_i))}{\partial \mathbf{Y}_i}\right)^\mathsf{T} \Lambda_i'^{-1} \frac{\partial \boldsymbol{\eta}(\mathbf{L}'_i, \mathbf{T}(\mathbf{Y}_i))}{\mathbf{Y}_i}.$$

Proof. By back-propagation of covariance, the MLE of $\mathbf{q}$ has covariance [9]

$$\text{cov}(\mathbf{q}) = \left(J_g^\mathsf{T} \Sigma_g^{-1} J_g\right)^{-1}, \tag{20}$$

where

$$J_g = \frac{\partial g}{\partial \mathbf{q}} = \left[\begin{array}{c|c} J_g^A & J_g^B \\ \hline J_g^C & J_g^D \end{array}\right]_{12m \times 6+6m}$$

$$= \left[\begin{array}{c|ccc} \mathbf{0} & \frac{\partial \boldsymbol{\eta}(\mathbf{L}_1, \mathbf{Y}_1)}{\partial \mathbf{Y}_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \frac{\partial \boldsymbol{\eta}(\mathbf{L}_m, \mathbf{Y}_m)}{\partial \mathbf{Y}_m} \\ \hline \frac{\partial \boldsymbol{\eta}(\mathbf{L}'_1, \mathbf{T}(\mathbf{Y}_1))}{\partial \boldsymbol{\xi}} & \frac{\partial \boldsymbol{\eta}(\mathbf{L}'_1, \mathbf{T}(\mathbf{Y}_1))}{\partial \mathbf{Y}_1} & \mathbf{0} & \mathbf{0} \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \frac{\partial \boldsymbol{\eta}(\mathbf{L}'_n, \mathbf{T}(\mathbf{Y}_n))}{\partial \boldsymbol{\xi}} & \mathbf{0} & \mathbf{0} & \frac{\partial \boldsymbol{\eta}(\mathbf{L}'_m, \mathbf{T}(\mathbf{Y}_m))}{\partial \mathbf{Y}_m} \end{array}\right]$$

With $\Sigma_g^{-1} = diag\left(\Lambda_1^{-1}, \cdots, \Lambda_m^{-1}, \Lambda_1'^{-1} \cdots, \Lambda_m'^{-1}\right)$, we derive $J_g \Sigma_g^{-1} J_g = \left[\begin{array}{c|c} H_g^A & H_g^B \\ \hline {H_g^B}^\mathsf{T} & H_g^D \end{array}\right]$. Performing blockwise matrix inversion on $J_g \Sigma_g^{-1} J_g$ yields (19). □

5.2.3 Motion estimation using points and lines

Now we are ready to fuse points and lines for the MLE of motion. Given $\{\mathbf{P}_i \leftrightarrow \mathbf{P}'_i, i = 1, \cdots, n\}$ and $\{\mathbf{L}_i \leftrightarrow \mathbf{L}'_i, i = 1, \cdots, m\}$, we formulate an MLE problem that jointly estimates the rigid motion, point landmarks and line landmarks. The parameter vector to be estimated is defined as

$$\mathbf{r} = \left[\boldsymbol{\xi}^\mathsf{T}, \mathbf{X}_1^\mathsf{T}, \cdots, \mathbf{X}_n^\mathsf{T}, \mathbf{Y}_1^\mathsf{T}, \cdots, \mathbf{Y}_m^\mathsf{T}\right]^\mathsf{T}.$$

Define a measurement error function

$$f(\mathbf{r}) = \begin{bmatrix} \mathbf{f_p} \\ \mathbf{f_l} \end{bmatrix}, \tag{21}$$

where $\mathbf{f_p} = \begin{bmatrix} \mathbf{X}_1 - \mathbf{P}_1 \\ \vdots \\ \mathbf{X}_n - \mathbf{P}_n \\ \mathbf{T}(\mathbf{X}_1) - \mathbf{P}'_1 \\ \vdots \\ \mathbf{T}(\mathbf{X}_n) - \mathbf{P}'_n \end{bmatrix}$ and $\mathbf{f_l} = \begin{bmatrix} \boldsymbol{\eta}(\mathbf{L}_1, \mathbf{Y}_1) \\ \vdots \\ \boldsymbol{\eta}(\mathbf{L}_m, \mathbf{Y}_m) \\ \boldsymbol{\eta}(\mathbf{L}'_1, \mathbf{T}(\mathbf{Y}_1)) \\ \vdots \\ \boldsymbol{\eta}(\mathbf{L}'_m, \mathbf{T}(\mathbf{Y}_m)) \end{bmatrix}$. The MLE of $\mathbf{r}$ is obtained by solving the following problem

$$\min_{\mathbf{r}} f(\mathbf{r})^\mathsf{T} \Sigma_f^{-1} f(\mathbf{r}), \tag{22}$$

where $\Sigma_f = diag(\Sigma_h, \Sigma_g)$.

Lemma 3. *Under Gaussian noise assumption, the MLE of rigid transformation $\boldsymbol{\xi}$ based on both point and line features obtained by (22) has covariance*

$$C_H = \left(H_h^A + H_g^A - H_h^B {H_h^D}^{-1} {H_h^B}^\mathsf{T} - H_g^B {H_g^D}^{-1} {H_g^B}^\mathsf{T}\right)^{-1}. \tag{23}$$

Proof. By back-propagation of covariance, the MLE of $\mathbf{r}$ has covariance [9]

$$\text{cov}(\mathbf{r}) = \left(J_f^\mathsf{T} \Sigma_f^{-1} J_f\right)^{-1} \tag{24}$$

where $J_f = \frac{\partial f}{\partial \mathbf{r}} = \left[\begin{array}{ccc} J_h^A & J_h^B & \mathbf{0} \\ J_h^C & J_h^D & \mathbf{0} \\ \hline J_g^A & \mathbf{0} & J_g^B \\ J_g^C & \mathbf{0} & J_g^D \end{array}\right]$. With $\Sigma_f^{-1} = diag\left(\Sigma_h^{-1}, \Sigma_g^{-1}\right)$, we derive $J_f^\mathsf{T} \Sigma_f^{-1} J_f = \left[\begin{array}{c|cc} H_h^A + H_g^A & H_h^B & H_g^B \\ \hline {H_h^B}^\mathsf{T} & H_h^D & \mathbf{0} \\ {H_g^B}^\mathsf{T} & \mathbf{0} & H_g^D \end{array}\right]$. Performing blockwise matrix inversion on $J_f^\mathsf{T} \Sigma_f^{-1} J_f$ yields (23).

□

With Lemmas 1, 2 and 3 introduced, we are ready to present the following theorem that justifies the benefit of fusing point and line features for RGB-D odometry.

Theorem 1. *Under Gaussian noise assumption, fusing points and lines produces smaller uncertainty in the MLE of pairwise motion than using each feature alone. Specifically, for the MLE covariances, C_P, C_L and C_H, obtained from using points, lines, and points plus lines respectively, we have*

$$\lambda_i(C_H) < \lambda_i(C_P),\ \lambda_i(C_H) < \lambda_i(C_L), 1 \le i \le 6 \tag{25}$$

where $\lambda_i(\cdot)$ denotes the i-th largest eigenvalue.

Proof. Let us write $M_1 \succ M_2$ if matrices M_1 and M_2 are real symmetric and $M_1 - M_2$ is positive definite.

By comparing (23) with (14) and (19), we find $C_H = \left(C_P^{-1} + C_L^{-1}\right)^{-1}$, which is equivalent to

$$C_H^{-1} = C_P^{-1} + C_L^{-1}. \tag{26}$$

It holds that $C_P \succ 0, C_L \succ 0$ since they are both covariance matrices, which further implies $C_P^{-1} \succ 0, C_L^{-1} \succ 0$.

As a result, we have

$$C_H^{-1} - C_P^{-1} = C_L^{-1} \succ 0 \text{ and } C_H^{-1} - C_L^{-1} = C_P^{-1} \succ 0,$$

which means $C_H^{-1} \succ C_P^{-1}$ and $C_H^{-1} \succ C_L^{-1}$. According to Theorem 7.7.3 in [11],

$$C_H^{-1} \succ C_P^{-1} \Leftrightarrow C_H \prec C_P, \text{ and } C_H^{-1} \succ C_L^{-1} \Leftrightarrow C_H \prec C_L,$$

which further leads to (25) per Corollary 7.7.4 in [11]. □

6. Experiments

Our system is implemented based on the software of [8], and named Point and Line based Visual Odometry (PLVO) [16]. The average processing time is 0.31s per frame (640×480) on a desktop with Intel Xeon E5-2609 CPU. We evaluate PLVO under both varying and constant lighting, and compare it with the following state-of-the-art algorithms:

- Kpoint: a representative keypoint based visual SLAM algorithm [8], open source software, referred to as Kpoint here.
- DVO: a recent dense visual SLAM method [12], open source software.
- Edge: the latest edge-based RGB-D method [5], referred to as Edge here. Edge is only compared on public dataset because it is not open source.

We start with the evaluation under varying lighting.

6.1. Test Under Varying Lighting

To evaluate PLVO under real-world scenarios, we record RGB-D data at 30 FPS by hand-holding a Kinect and walking in typical indoor environments, including corridors,

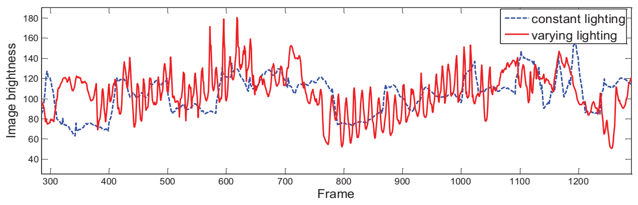


Figure 3: Example of image brightness change over time under constant/varying lighting (from Corridor-C). Here image brightness means the average intensity of an image.

staircases and halls. The trajectory lengths, listed in Table 1, 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). Figure 3 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 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 in this test. Table 1 shows the test results, where we use bold font to indicate the best result in each row. As Kpoint allows using various point detectors, we report the best result for each sequence obtained by respectively applying SIFT [15], SURF and ORB [27]. From Table 1, we see that PLVO achieves the least TED on the majority of sequences, under both constant and varying lighting conditions. This clearly demonstrates the accuracy advantage of PLVO, as well as its robustness against lighting variations.

Table 1: TED (IN METERS)

Site (travel distance)	Lighting	Kpoint	DVO	PLVO
Corridor-A (82 m)	constant	**4.36**	7.10	7.50
	varying	16.68	15.41	**12.20**
Corridor-B (77 m)	constant	8.25	7.56	**5.28**
	varying	12.75	12.96	**5.15**
Corridor-C (86 m)	constant	6.53	6.12	**5.70**
	varying	7.30	5.93	**3.46**
Staircase-A (52 m)	constant	4.04	2.26	**2.13**
	varying	4.47	3.17	**2.41**
Staircase-B (45 m)	constant	5.77	**1.72**	4.50
	varying	**3.12**	3.35	6.41
Staircase-C (41 m)	constant	4.51	13.87	**2.74**
	varying	8.79	16.00	**1.86**
Entry-Hall (54 m)	constant	1.53	**1.31**	1.63
	varying	3.78	6.59	**3.70**
Auditorium (53 m)	constant	5.78	2.39	**1.86**
	varying	6.74	10.66	**4.44**
Classroom (45 m)	constant	2.47	3.48	**1.93**
	varying	2.58	4.73	**2.16**

6.2. Test on TUM Dataset Under Constant Lighting

We also evaluate our method under constant lighting using the TUM FR1 dataset [32], 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 [32]. For a given interval Δ, the RPE at time instant i is defined as

$$\mathbf{E}_i := \left(\mathsf{Q}_i^{-1}\mathsf{Q}_{i+\Delta}\right)^{-1}\left(\mathsf{P}_i^{-1}\mathsf{P}_{i+\Delta}\right), \qquad (27)$$

where $\mathsf{Q}_i \in \mathrm{SE}(3)$ and $\mathsf{P}_i \in \mathrm{SE}(3)$ 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.

Table 2 contains two comparison results. On the left part, we compare PLVO with Kpoint and Edge, where the RPE is computed with $\Delta = 1$ frame in (27). The errors of Kpoint are computed using their published resulting trajectories [1]. The errors of Edge are directly excerpted from [5]. For each sequence, the first and second rows represent the translational and rotational errors, respectively. It shows that PLVO outperforms its counterparts. We compute an average error over all sequences weighted by their frame numbers. Our method achieves the smallest average errors. Specifically, the average translational and rotational errors of PLVO are 42.5% and 28.3% smaller than the second best one (Edge), respectively.

Table 2: RMSE OF RPE ON TUM FR1 SEQUENCES

Seq. (#Frame)	Kpoint	Edge	PLVO	PLVO	DVO
	error per frame			error per second	
360 (745)	13.8 mm 0.63 deg	**11.2** mm 0.55 deg	**11.2** mm **0.45** deg	**84** mm	119 mm
desk (575)	11.7 mm 0.73 deg	**8.6** mm 0.70 deg	10.8 mm **0.60** deg	39 mm	**30** mm
desk2 (614)	17.6 mm 1.07 deg	**8.9** mm 0.7 deg	11.5 mm **0.64** deg	**54** mm	55 mm
floor (1214)	3.7 mm 0.29 deg	15.7 mm 0.47 deg	**3.5** mm **0.28** deg	**24** mm	90 mm
plant (1112)	20.7 mm 1.25 deg	6.9 mm 0.49 deg	**5.1** mm **0.34** deg	**24** mm	36 mm
room (1332)	13.7 mm 0.63 deg	6.2 mm 0.48 deg	**5.3** mm **0.36** deg	49 mm	**48** mm
rpy (687)	12.1 mm 0.91 deg	**7.2** mm 0.67 deg	9.1 mm **0.63** deg	52 mm	**43** mm
teddy (1395)	25.4 mm 1.45 deg	36.5 mm 0.92 deg	**11.5** mm **0.47** deg	**50** mm	67 mm
xyz (788)	5.8 mm **0.35** deg	**4.7** mm 0.41 deg	5.3 mm **0.35** deg	**22** mm	24 mm
weighted mean	14.4 mm 0.83 deg	13.4 mm 0.60 deg	**7.7** mm **0.43** deg	**43** mm	58 mm

We compare PLVO with DVO separately in the right part of Table 2 because only translational errors are reported in [12] with its unit being m/s, i.e. $\Delta = 1$ s in (27). PLVO produces an average translational error which is 34.9% smaller than DVO, as highlighted in Abstract.

7. Conclusion and future work

To improve visual odometry robustness, we proposed an RGB-D camera based method by fusing point and line features. We proved that fusing these two types of features produced smaller uncertainty in the MLE of relative motion than using either feature type alone. Our method was evaluated on real-world data in experiments. We compared its performance with state-of-the-art methods Kpoint, Edge and DVO, under both constant and varying lighting. Our method exhibited both superior robustness to lighting change and better accuracy in either settings. In the future, we will investigate how to enhance the system by using pose graph optimization and by detecting loop closure. Fusing inertial sensors with an RGB-D camera is another direction to explore.

References

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

[2] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (5):698–700, 1987.

[3] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, pages 404–417. Springer, 2006.

[4] 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)*, pages 2769–2776, 2014.

[5] 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)*, pages 1568–1575, 2013.

[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*, 29(6):1052–1067, June 2007.

[7] E. Eade and T. Drummond. Edge landmarks in monocular SLAM. *Image and Vision Computing*, 27(5):588 – 596, 2009.

[8] 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)*, pages 1691–1696, 2012.

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

[10] 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*, 31(5):647–663, 2012.

[11] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge university press, 2012.

[12] 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)*, pages 2100–2106, 2013.

[13] 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)*, pages 225–234, 2007.

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

[15] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(4):91–110, November 2004.

[16] Y. Lu and D. Song. RGB-D odometry using Point and line Features. `http://telerobot.cs.tamu.edu/MFG/rgbd/plvo`, 2015.

[17] Y. Lu and D. Song. Robustness to lighting variations: An RGB-D indoor visual odometry using line segments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page [Accepted], 2015.

[18] Y. Lu and D. Song. Visual navigation using heterogeneous landmarks and unsupervised geometric constraints. *IEEE Transactions on Robotics (T-RO)*, 31(3):736–749, June 2015.

[19] 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)*, pages 162–167, 2013.

[20] 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)*, pages 1540–1545, 2014.

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

[22] 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)*, pages 127–136, 2011.

[23] 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)*, pages 2320–2327, 2011.

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

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

[26] 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.

[27] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2564–2571, 2011.

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

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

[30] 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*, pages 719–722, 2011.

[31] H. Strasdat, J. M. Montiel, and A. J. Davison. Visual SLAM: Why filter? *Image and Vision Computing*, 30(2):65–77, 2012.

[32] 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)*, pages 573–580, 2012.

[33] 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)*, pages 5182–5189, 2013.

[34] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991.

[35] 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*, 32(4):722–732, April 2010.

[36] Z. Wang, F. Wu, and Z. Hu. MSLD: A robust descriptor for line matching. *Pattern Recognition*, 42(5):941–953, 2009.

[37] 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)*, pages 5724–5731, 2013.

[38] Z. Zhang and O. D. Faugeras. Determining motion from 3D line segment matches: A comparative study. *Image and Vision Computing*, 9(1):10–19, 1991.