Elaborate Monocular Point and Line SLAM with Robust Initialization

Sang Jun Lee
School of Computer Science and Electrical Engineering
Handong Global University, Korea
eowjd4@naver.com

Sung Soo Hwang
School of Computer Science and Electrical Engineering
Handong Global University, Korea
sshwang@handong.edu

Abstract

This paper presents a monocular indirect SLAM system which performs robust initialization and accurate localization. For initialization, we utilize a matrix factorization-based method. Matrix factorization-based methods require that extracted feature points must be tracked in all used frames. Since consistent tracking is difficult in challenging environments, a geometric interpolation that utilizes epipolar geometry is proposed. For localization, 3D lines are utilized. We propose the use of Plücker line coordinates to represent geometric information of lines. We also propose orthonormal representation of Plücker line coordinates and Jacobians of lines for better optimization. Experimental results show that the proposed initialization generates consistent and robust map in linear time with fast convergence even in challenging scenes. And localization using proposed line representations is faster, more accurate and memory efficient than other state-of-the-art methods.

1. Introduction

Nowadays, the interests of visual SLAM(Simultaneously Localization and Mapping) has been increased since it has been used for augmented reality, autonomous driving car, and robotics as an important component. Moreover, the monocular camera has been widely considered for the visual SLAM due to not only its inexpensiveness but also a widely used equipment in many industries. However, monocular visual SLAM has several drawbacks such as scale drift, pure rotation, etc., which come from the use of single image to operate the system [1].

In particular, map initialization in monocular visual SLAM is more challenging than sensor-based SLAMs. Since map initialization is highly related with the system performance, special efforts should be made on robust initialization. Conventional initialization methods in indirect (feature-based) monocular visual SLAM utilize feature points to optimize the system by minimizing geometric errors [10, 14]. These methods find corresponding feature points between two frames, calculate relative camera pose, and then reconstruct 3D landmarks using them. However, the estimated camera pose calculated from the estimated fundamental matrix has multiple solutions, or the one from homography is specified for movement on planar scene only. Model selection between fundamental matrix and homography is also tricky as discussed in [2, 10]. In addition, highly restricted criterions such as detecting low-parallax cases and twofold planar ambiguity [3] causes initialization to fail. Moreover, frames are simply abandoned when the initialization is failed which may slow down convergence.

In contrast, direct (pixel-based) monocular visual SLAM systems that use pixel-wise matching for minimizing photometric errors generate random initial map and update it using consecutively associated data. However, these methods may generate unstable result as discussed in [4]. Besides, Direct-based methods are accurate only if the input images are rectified by photometric calibration [5, 6].

Currently, Tang et al. [7] suggested robust initialization based on rank-1 matrix factorization. This method generates initial map free from model selection problem. Furthermore, this method guarantees fast convergence with linear time by optimizing all involved cameras and points simultaneously. However, to operate matrix factorization, all feature points that are used for matrix factorization must be tracked in all using frames. To address this problem, Tang et al. used KLT tracking-based system to track feature points. However, this method is highly relied on the performance of KLT tracking [8]. And it may be failed by illumination change and large baseline movement as discussed in [9]. Moreover, by the nature of KLT tracking, lost features are hardly recovered. Therefore, KLT-based SLAM only utilizes RANSAC-based n-points algorithms [25, 26, 27] for pose estimation that is less accurate than state-of-the-art indirect SLAM systems [10, 11].

Another critical issue in monocular SLAM is localization. For robust localization, current methods utilize line and point features simultaneously [12, 20]. Line is a geometric primitive that has dual relation with point, thus it produces valuable geometric information as important as point even though the representation is different. Pumarola, et al. proposed the state-of-the-art method using points and lines [12]. This method represents a line segment with its two endpoints, and reconstructs a 3D line by triangulating each

endpoint. All optimization of re-projection error on 3D line is operated as optimizing two 3D endpoints that are over parameterized. Even though the endpoints representation of line is also a good approach, this cannot reliably reconstruct 3D lines due to endpoints shifting. Assume there is a detected line segment lying on a projected line from a 3D straight line. Even if two endpoints of the line segment are shifted on the projected line, line's internal coefficients are same yet the positions are different. This causes a factor of deficiency of the system.

In this paper, we propose a robust initialization and localization method for monocular indirect visual SLAM. For robust initialization, we utilize rank-1 factorization. In feature-based indirect method, features are hardly tracked in all frames, so in order to handle this problem, we propose a geometric interpolation utilizing a computational trick[1, 36]. The geometric interpolation is based on well-known epipolar constraint to make all features to be tracked in all frames, and the computational trick is proposed for efficient computation of essential matrices for the geometric interpolation.

For accurate localization, the proposed method utilizes Plücker line representation which represents 3D lines geometrically well in homogeneous coordinates. Plücker line representation has less parameters than endpoints representation, and there is no shifting situation. Thus, this representation is computationally cheaper and more geometrically robust than endpoints representation. Even though Plücker line representation has been utilized in other SLAMs [18-23], to our best knowledge, this is the first approach to use Plücker line representation in monocular indirect SLAM. In addition, we employ their orthonormal representation in which minimal parameters are retained. Based on this parameterization, Jacobians of lines are analytically calculated for pose and line graph optimization and they update line by decoupling lines without any gauge-freedoms. It leads the line optimization to be geometrically robust, efficient, and fast. Finally, we suggest solving degeneracy of line reconstruction occurred from two-view reconstruction by n-view reconstruction verified by the proposed initialization.

This paper is organized as follows. Section 2 and 3 demonstrate details of the factorization-based initialization and point-line SLAM using Plücker coordinates, respectively. Then, we explain implementation details in Section 4. Section 5 evaluates the experiments, and Section 6 concludes this paper.

2. Robust initialization

Figure 1 shows the outline of the proposed rank-1 factorization-based initialization. The proposed initialization method associates subsequent *m* frames from initial frame and *n* feature points that are tracked in all *m* frames to make a matrix A as shown in Figure 1. (b).

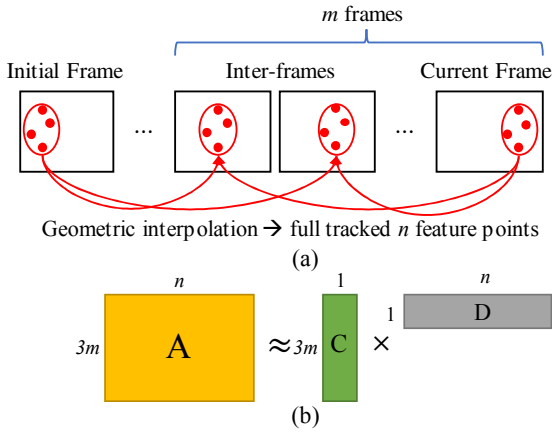


Figure 1: Overview of rank-1 factorization-based initialization. (a) Data association using geometric interpolation, (b) Rank-1 factorization using associated data from (a).

However, feature matching by using its descriptor may not be tracked in all frames. Therefore, the proposed geometric interpolation obtains *n* feature points as being tracked in all *m* frames as shown in Figure 1. (a). We demonstrate more details how to construct matrix A in Section 3.1 and how the geometric interpolation is performed in Section 3.2.

For all notations in this paper, we use bold-type for a vector regardless of uppercase and lowercase, uppercase without bold-type is used for matrix, and italic-type is used for a scalar or index. All points are in homogeneous coordinates, i.e., $\mathbf{x} = (\bar{\mathbf{x}}^{\mathrm{T}}, 1)^{\mathrm{T}} \in \mathbb{R}^3, \mathbf{X} = (\bar{\mathbf{X}}^{\mathrm{T}}, 1)^{\mathrm{T}} \in \mathbb{R}^4$ where upper-bar at $\bar{\mathbf{x}}$ indicates inhomogeneous coordinates. In addition, line and plane are basically represented in homogeneous coordinates.

2.1. Point-camera constraints & factorization

To factorize a matrix into all camera poses and 3D points simultaneously, all entities in a matrix must be filled by point-camera constraints widely used in SfM [7, 16, 17] illustrated in Figure 2.

Let F_i is *i*-th frame and $\mathbf{c}_i$ is the camera position of F_i, then F_0 is initial frame and $\mathbf{c}_0$ is the camera position of F_0 located at origin. $\mathbf{p}_k$ is a *k*-th 3D point with inverse depth d_k viewed from F_0. Because d_k is unknown, d_k is set by one to make derivation simple and it is generalized later. Then, $\mathbf{p}_k$ becomes the ray viewed from $\mathbf{c}_0$ computed by normalizing the corresponding feature point in F_0.

Assuming there is no noise, the $\mathbf{c}_i$ becomes

$$\mathbf{c}_i = s_{0i}\mathbf{t}_i = \mathbf{p}_k - s_{ik}\mathbf{p}_{ik}, \tag{1}$$

where s_{0i} and s_{ik} are scale coefficients for that constraints, $\mathbf{p}_{ik}$ is $\mathrm{R}_i^{\mathrm{T}}\mathbf{p}_k$, and the translation $\mathbf{t}_i$ and rotation R_i are camera pose of $\mathbf{c}_i$. The camera pose can be calculated by eight-point algorithm [25] or five-point algorithm for rotation [26] and two-point algorithm for translation [27].

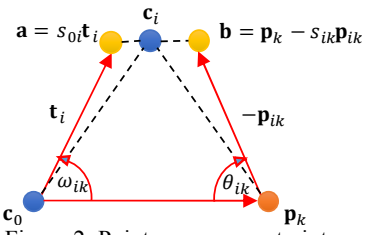


Figure 2: Point-camera constraints.

However, noise always exists in real data, $\mathbf{c}_i$ is approximated by the midpoint of two representations as:

$$\mathbf{c}_i \approx \frac{1}{2}(\mathbf{a}+\mathbf{b}) = \frac{1}{2}(s_{0i}\mathbf{t}_i + \mathbf{p}_k - s_{ik}\mathbf{p}_{ik}), \quad (2)$$

and s_{0i} and s_{ik} can be estimated by solving the simple matrix equation as:

$$\begin{bmatrix}\mathbf{t}_i^{\mathrm{T}} \\ \mathbf{p}_{ik}^{\mathrm{T}}\end{bmatrix}(\mathbf{a}-\mathbf{b}) = \mathbf{0}_{2\times 1}. \quad (3)$$

Then, we now rewrite Equation (2) replacing $\mathbf{t}_i$ to $\mathrm{R}(\omega_{ik})\cdot\mathbf{p}_k$ that is the rotation matrix around the axis $\mathbf{p}_k \times \mathbf{t}_i$ for an angle ω_{ik}, and $\mathbf{p}_{ik}$ to $\mathrm{R}(\theta_{ik})\mathbf{p}_k$ that is the rotation matrix around the axis $\mathbf{p}_k \times \mathbf{p}_{ik}$ for an angle θ_{ik}, as:

$$\mathbf{c}_i \approx \frac{1}{2}(s_{0i}\mathrm{R}(\omega_{ik})\mathbf{p}_k + \mathbf{p}_k - s_{ik}\mathrm{R}(\theta_{ik})\mathbf{p}_k) = \mathbf{v}_{ik}, \quad (4)$$

where $\mathbf{A}_{ik} = 1/2(s_{0i}\mathrm{R}(\omega_{ik})\mathbf{p}_k + \mathbf{p}_k - s_{ik}\mathrm{R}(\theta_{ik})\mathbf{p}_k)$ is a known vector. By generalizing $\mathbf{p}_k$ in Equation (4) to $1/d_k \cdot \mathbf{p}_k$ by using inverse depth, we finally get the equation $\mathbf{c}_i \approx 1/d_k \cdot \mathbf{v}_{ik}$, and we solve

$$\arg\min_{\substack{\mathbf{c}_{i=1\cdots m} \\ \mathbf{d}_{k=1\cdots n}}} \sum_{i=1}^{m}\sum_{k=1}^{n} \| \mathbf{c}_i d_k - \mathbf{A}_{ik} \|_2^2, \quad (5)$$

where m is the number of subsequent frames that are used for matrix factorization, and n is the number of key points that are tracked in all m frames, $\| \mathbf{c}_i d_k - \mathbf{A}_{ik} \|_2^2$ is a reweighted geometric error suggested in [7]. We solve Equation (5) by rank-1 factorization because depth has rank one. Equation (5) can be solved by using SVD (Singular Value Decomposition) with Lanczos algorithm [28] that is an adaption of power methods to find one eigen vector corresponding to the most useful one eigenvalue. Therefore, a known matrix $\mathrm{A}_{3m\times n}$ is decomposed into camera position matrix $\mathrm{C}_{3m\times 1}$ and depth matrix $\mathrm{D}_{1\times n}$ while reconstructing all cameras and depths simultaneously.

2.2. Matrix completion by geometric interpolation

In order to apply factorization-based initialization to indirect SLAM system, we conduct geometric interpolation for matrix completion. For precise estimation of location of un-matched feature points, we utilize epipolar geometry to estimate appropriate location using known camera poses. We first demonstrate how to interpolate the location, and then show how to conduct geometric interpolation.

As shown in Figure 3, given all the camera poses as initial frame $[\mathrm{I}|\mathbf{0}]$, current frame $[\mathrm{R}|\mathbf{t}]_c$ and subsequent inter-frames $[\mathrm{R}|\mathbf{t}]_{i=1\cdots c-1}$ with a matching pair $\bar{\mathbf{x}}_0 \leftrightarrow \bar{\mathbf{x}}_c$ on initial frame and current frame, the interpolated point $\bar{\mathbf{x}}_i$ in an inter-frame is estimated by intersecting two epipolar lines $\mathbf{l}_0, \mathbf{l}_c$ as $\mathbf{x}_i = \mathbf{l}_0 \times \mathbf{l}_c$. The line $\mathbf{l}_0$ is calculated forward from initial frame as $\mathbf{l}_0 = \mathrm{E}_0^i\mathbf{x}_0$ where E_0^i is the essential matrix constructed by the relative pose $[\mathrm{R}|\mathbf{t}]_i$ from initial frame to i-th inter-frame. The line $\mathbf{l}_c$ is calculated backward from current frame $\mathbf{l}_c = \mathrm{E}_c^i\mathbf{x}_c$ where E_c^i is constructed by relative pose $[\mathrm{R}|\mathbf{t}]_i$ $[\mathrm{R}|\mathbf{t}]_c^{-1}$ from current frame to i-th inter-frame. In the Figure 3, '$\rightarrow$' indicates the relation of constructing relative camera pose to essential matrix formed as $[\mathbf{t}]_\times\mathrm{R}$ where $[\cdot]_\times$ indicates the skew-symmetric matrix of a vector.

When the geometric interpolation is operated, it calculates the essential matrix from current frame to an inter frame. It needs the inverse matrix of the current frame's camera pose that leads to over computation in interpolation for all features to be tracked in all frames. To prevent the calculation of the inverse matrix, we utilize the face that frames of SLAM system have been captured sequentially. The idea is to pre-calculate some of valuable information of each frame to avoid calculation of inverse. We notice that the essential matrix from current frame to inter frames can be obtained by incrementally multiplying the essential matrices between the inter frame and its next inter frame in the backward direction from current frame to initial frame.

To this end, we store E_0^c and relative pose from previous frame $[\mathrm{R}|\mathbf{t}]_{c-1}^c = [\mathrm{R}|\mathbf{t}]_{c-1}^{-1}[\mathrm{R}|\mathbf{t}]_c$ with its inverse $[\mathrm{R}|\mathbf{t}]_c^{c-1}$ in every time. Then, it is easy to calculate essential matrix from current frame to inter-frames as like as chain rule when the interpolation is operated. For an example, a point in i-th frame is interpolated by backward process from current frame calculating the position using two essential matrix one from initial frame E_0^i constructed by own camera pose $[\mathrm{R}|\mathbf{t}]_i$, and another one from current frame $\mathrm{E}_c^i = \mathrm{E}_{i+1}^i\,\mathrm{E}_{i+2}^{i+1}\cdots\mathrm{E}_{c-1}^{c-2}\mathrm{E}_c^{c-1}$, which is obtained by incremental multiplication of pre-stored inverse matrix as shown in Figure 4. Utilizing this method, after subsequent m frames are stacked, each and all matched feature points between initial frame and current frame is interpolated, and matrix factorization is operated.

The outliers of points that are used for matrix completion are rejected by symmetric epipolar distance [1]. We use 3.84 based on $\mathcal{X}^2$ distribution test at 95% for the threshold of symmetric epipolar distance. Furthermore, to avoid degeneracy that is occurred when the movement of features

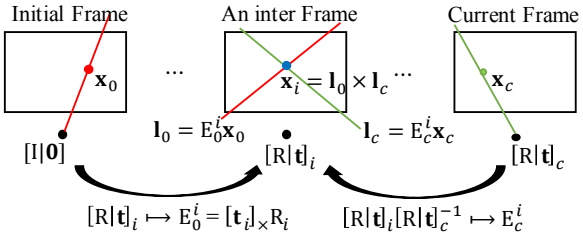


Figure 3: Overview of geometric interpolation. Using two points in initial frame and current frame estimates interpolated points in inter-frames.

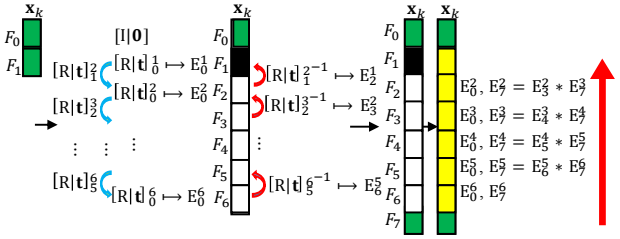


Figure 4: Efficient operation for geometric interpolation. $\mathbf{x}_k$ is a feature point matched between initial frame and current frame.

lies on the epipolar plane in geometric interpolation, we reject the features from matrix completion if an interpolated feature lies on epipolar plane calculated from initial frame and current frame have the angle between -5 and 5 degree.

3. Line-based localization

In this Section, we demonstrate the proposed line-based localization. We utilize Plücker coordinates and their orthonormal representation to reconstruct and represent 3D lines. We introduce the concept of those representations in homogeneous coordinates in Section 3.1, and define the re-projection error and the cost function with regard to those representations to optimize pose and line graph for the system in Section 3.2. Then, we explain line reconstruction methods solving several degeneracy cases in Section 3.3.

3.1. Plücker & orthonormal representation

3D line represented by Plücker coordinates consists of two 3D points $\mathbf{X}_1 \sim (\bar{\mathbf{X}}_1^T | r_1)^{\mathrm{T}}$ and $\mathbf{X}_2 \sim (\bar{\mathbf{X}}_2^T | r_2)^{\mathrm{T}}$ according to the way in [34] as following Equation (6):

$$\mathbf{L} = \begin{bmatrix} \mathbf{m} \\ \mathbf{d} \end{bmatrix} \in \mathbb{P}^5 \subset \mathbb{R}^6, \tag{6}$$

where $\mathbf{d} = r_1 \bar{\mathbf{X}}_2 - r_2 \bar{\mathbf{X}}_1$ is direction vector, and $\mathbf{m} = \mathbf{d} \times \bar{\mathbf{X}}_1$ is moment vector that indicates normal of the line.

Plücker coordinates described in [1] represent line with 5 d. o. f (degree of freedom) in homogeneous coordinates satisfying Klein quadric constraints $\mathbf{m}^{\mathrm{T}}\mathbf{d} = 0$. In addition, Plücker coordinates also can apply linear projection in homogeneous coordinates. When K is camera intrinsic matrix with squared pixel, and T_{cw} is extrinsic matrix for a 3D point as:

$$\mathrm{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \mathrm{T}_{cw} = \begin{bmatrix} \mathrm{R}_{cw} & \mathbf{t}_{cw} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix}. \tag{7}$$

Then, intrinsic matrix $\mathcal{K}$ and extrinsic matrix $\mathcal{H}$ for a 3D line are as follows:

$$\mathcal{K} = \begin{bmatrix} f_y & 0 & 0 \\ 0 & f_x & 0 \\ -f_y c_x & -f_x c_y & f_x f_y \end{bmatrix}, \tag{8}$$

$$\mathcal{H}_{cw} = \begin{bmatrix} \mathrm{R}_{cw} & [\mathbf{t}_{cw}]_\times \mathrm{R}_{cw} \\ 0_{3\times 3} & \mathrm{R}_{cw} \end{bmatrix}. \tag{9}$$

However, Plücker coordinates also have two gauge-freedom itself, so orthonormal representation of Plücker coordinates is suggested to parameterize it to minimal four parameters [29].

Any Plücker coordinates can be represented by orthonormal representation $(\mathrm{U}, \mathrm{W}) \in SO(3) \times SO(2)$ where $SO(\cdot)$ is special orthogonal groups of Lie algebra [30] as:

$$\mathrm{U} = \left[\frac{\mathbf{m}}{\| \mathbf{m} \|} \quad \frac{\mathbf{d}}{\| \mathbf{d} \|} \quad \frac{\mathbf{m} \times \mathbf{d}}{\| \mathbf{m} \times \mathbf{d} \|} \right], \tag{10}$$

$$\mathrm{W} = \frac{1}{\| \mathbf{m} \| \| \mathbf{d} \|} \begin{bmatrix} \| \mathbf{m} \| & \| \mathbf{d} \| \\ -\| \mathbf{d} \| & \| \mathbf{m} \| \end{bmatrix}. \tag{11}$$

U and W can be updated as $\mathrm{U} \leftarrow \mathrm{UR}(\boldsymbol{\theta})$ where $\boldsymbol{\theta} \in \mathbb{R}^3$ and $\mathrm{W} \leftarrow \mathrm{WR}(\theta)$ where $\theta \in \mathbb{R}$, respectively, and rotation matrix $\mathrm{R}(\boldsymbol{\theta})$ is represented by exponential map. Therefore, $\boldsymbol{\delta}_{\boldsymbol{\theta}} = [\boldsymbol{\theta}^{\mathrm{T}}, \theta] \in \mathbb{R}^4$ is the four minimal parameters to update orthonormal representation. The Plücker coordinates can be recovered from orthonormal representation as $\mathbf{L}^{\mathrm{T}} \sim \left(w_{11}\mathbf{u}_1^{\mathrm{T}}, w_{21}\mathbf{u}_2^{\mathrm{T}} \right)$, where w_{ij} is an element in W, and $\mathbf{u}_i$ is the i-th column of U. The orthonormal representation is used for the optimization of the re-projection errors on line.

3.2. Point-line & pose graph optimization

Now, we define line re-projection error between estimated line and observed line segment for line and pose graph optimization.

Let $\mathbf{L}_w$ is a Plücker line in 3D-space, then Plücker line $\mathbf{L}_c$ in camera coordinates and line $\mathbf{l}_c$ in image space that projected by $\mathbf{L}_c$ are obtained as:

$$\mathbf{L}_c = \begin{bmatrix} \mathbf{m}_c \\ \mathbf{d}_c \end{bmatrix} = \mathcal{H}_{cw} \mathbf{L}_w \ , \mathbf{l}_c = \mathcal{K} \mathbf{m}_c \in \mathbb{R}^3. \tag{12}$$

Then, we define orthogonal distance between $\mathbf{l}_c$ and measured two endpoints $\mathbf{x}_s, \mathbf{x}_e$ from the observed line segment $\boldsymbol{z}$ as:

$$\mathrm{d}(\boldsymbol{z}, \mathbf{l}_c) = \left[\frac{\mathbf{x}_s^{\mathrm{T}} \mathbf{l}_c}{\sqrt{l_1^2 + l_2^2}}, \frac{\mathbf{x}_e^{\mathrm{T}} \mathbf{l}_c}{\sqrt{l_1^2 + l_2^2}} \right]^{\mathrm{T}}, \tag{13}$$

where $\mathrm{d}(\cdot)$ is the distance function.

The camera pose T_{kw}, the 3D point position $\mathbf{X}_{wi}$, and the position of 3D line $\mathbf{L}_{wj}$ are denoted as vertices in the graph model. In the graph model, a vertex of T_{kw} connects each $\mathbf{X}_{wi}$ and $\mathbf{L}_{wj}$ as two types of edges. Then, the re-projection errors in the edges are as follows:

$$Ep_{ki} = \mathbf{x}_{ki} - \pi(\mathrm{KT}_{kw}\bar{\mathbf{X}}_{wi}), El_{kj} = \mathrm{d}\big(\boldsymbol{z}_{kj}, \mathbf{l}_{kj}\big), \quad (14)$$

where Ep_{ki} and El_{kj} indicate the re-projection errors for point and line, respectively, $\mathbf{x}_{ki}$ is the measured position of feature point corresponding 3D point $\mathbf{X}_{wi}$ in image, $\pi(\cdot)$ represents inhomogeneous 2D points dividing by last element in homogeneous coordinates, and $\mathbf{l}_{kj} = \mathcal{K}\mathbf{m}_{kj}$, where $\mathbf{L}_{kj} = \begin{bmatrix}\mathbf{m}_{kj}^{\mathrm{T}} & \mathbf{d}_{kj}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} = \mathcal{H}_{kw}\mathbf{L}_{wj}$.

Therefore, the cost function C for point-line and pose graph optimization is constructed as:

$$C = \sum_{k,i} \rho(Ep_{ki}^{\mathrm{T}}\Sigma p_{ki}^{-1}Ep_{ki}) + \sum_{k,j} \rho(El_{kj}^{\mathrm{T}}\Sigma l_{kj}^{-1}El_{kj}), \quad (15)$$

where $\rho(\cdot)$ is robust Huber cost function, Σp_{ki}^{-1} and Σl_{kj}^{-1} are information matrix of point and line as their inverse covariance matrices.

To get Jacobians according to line re-projection error to optimize the cost function C as an iterative approach, we calculate analytic computed Jacobians for line parameters and camera poses. We skipped derivation about point, as it is well known. The Jacobians for line can be analytically calculated by chain rule to make derivation simple using following derivations. First, the partial derivative of re-projection of line $e_l = \mathrm{d}(\boldsymbol{z}, \mathbf{l})$ with respect to the line $\mathbf{l}$ is given by:

$$\frac{\partial e_l}{\partial \mathbf{l}} = \frac{1}{\sqrt{l_1^2 + l_2^2}}\begin{bmatrix} x_s - \dfrac{l_1\mathbf{x}_s\mathbf{l}}{\sqrt{l_1^2 + l_2^2}} & y_s - \dfrac{l_2\mathbf{x}_s\mathbf{l}}{\sqrt{l_1^2 + l_2^2}} & 1 \\ x_e - \dfrac{l_1\mathbf{x}_e\mathbf{l}}{\sqrt{l_1^2 + l_2^2}} & y_e - \dfrac{l_2\mathbf{x}_e\mathbf{l}}{\sqrt{l_1^2 + l_2^2}} & 1 \end{bmatrix}_{2\times 3}. \quad (16)$$

Then, partial derivatives of $\mathbf{l}$ with respect to $\mathbf{L}_{\mathrm{c}}$, and $\mathbf{L}_{\mathrm{c}}$ by $\mathbf{L}_{\mathrm{w}}$ are as follows:

$$\frac{\partial \mathbf{l}}{\partial \mathbf{L}_c} = \frac{\partial \mathcal{K}\mathbf{m}_c}{\partial \mathbf{L}_c} = [\mathcal{K} \quad 0_{3\times 3}]_{3\times 6}, \quad (17)$$

$$\frac{\partial \mathbf{L}_c}{\partial \mathbf{L}_w} = \frac{\partial \mathcal{H}_{cw}\mathbf{L}_w}{\partial \mathbf{L}_w} = \mathcal{H}_{cw}. \quad (18)$$

We use orthonormal representation to update minimal parameters of $\mathbf{L}_{\mathrm{w}}$, so we directly write Jacobian of $\mathbf{L}_{\mathrm{w}}$with respect to $\boldsymbol{\delta}_{\boldsymbol{\theta}}$ suggested in [29] :

$$\frac{\partial \mathbf{L}_w}{\partial \boldsymbol{\delta}_{\boldsymbol{\theta}}} = \begin{bmatrix} \mathbf{0}_{3\times 1} & -w_{11}\mathbf{u}_3 & w_{11}\mathbf{u}_2 & -w_{12}\mathbf{u}_1 \\ w_{12}\mathbf{u}_3 & \mathbf{0}_{3\times 1} & -w_{12}\mathbf{u}_1 & w_{11}\mathbf{u}_2 \end{bmatrix}_{6\times 4}. \quad (19)$$

For camera pose update, Jacobian matrix for camera pose in camera coordinates is given by:

$$\frac{\partial \mathbf{L}_c}{\partial \boldsymbol{\delta}_{\boldsymbol{\xi}}} = \begin{bmatrix} -[\mathrm{R}\mathbf{m}]_\times - [[\mathbf{t}]_\times \mathrm{R}\mathbf{d}]_\times & -[\mathrm{R}\mathbf{d}]_\times \\ -[\mathrm{R}\mathbf{d}]_\times & 0_{3\times 3} \end{bmatrix}_{6\times 6}, \quad (20)$$

where $\boldsymbol{\delta}_{\boldsymbol{\xi}}$ denotes the parameters of camera pose. The Equation (20) is derived by Zuo *et al.* in [20].

Finally, the complete Jacobians of re-projection error of line for line and camera pose are as follows respectively:

$$J_{\boldsymbol{\theta}} = \frac{\partial e_l}{\partial \boldsymbol{\delta}_{\boldsymbol{\theta}}} = \frac{\partial e_l}{\partial \mathbf{l}}\frac{\partial \mathbf{l}}{\partial \mathbf{L}_c}\frac{\partial \mathbf{L}_c}{\partial \mathbf{L}_w}\frac{\partial \mathbf{L}_w}{\partial \boldsymbol{\delta}_{\boldsymbol{\theta}}}, \quad (21)$$

$$J_{\boldsymbol{\xi}} = \frac{\partial e_l}{\partial \boldsymbol{\delta}_{\boldsymbol{\xi}}} = \frac{\partial e_l}{\partial \mathbf{l}}\frac{\partial \mathbf{l}}{\partial \mathbf{L}_c}\frac{\partial \mathbf{L}_c}{\partial \boldsymbol{\delta}_{\boldsymbol{\xi}}}. \quad (22)$$

All optimization with respect to line is conducted by the analytically computed Jacobians in iterative approaches.

3.3. Line reconstruction

Reconstruction of 3D Line using two-views is conducted by following steps. Given two camera projection matrices $\mathrm{P}_1, \mathrm{P}_2$ where $\mathrm{P}_i = \mathrm{KT}_i \in \mathbb{R}^{3\times 4}$, and matched each line segments $\boldsymbol{z}_1, \boldsymbol{z}_2$ in each camera image, where $\boldsymbol{z}$ includes two endpoints $\{\mathbf{x}_s, \mathbf{x}_e\}$. Constructing two planes $\boldsymbol{\pi}_1$, $\boldsymbol{\pi}_2$ as:

$$\boldsymbol{\pi}_1 = \mathbf{l}_1^{\mathrm{T}}\mathrm{P}_1, \boldsymbol{\pi}_2 = \mathbf{l}_2^{\mathrm{T}}\mathrm{P}_2, \quad (23)$$

where $\mathbf{l} = \mathbf{x}_s \times \mathbf{x}_e$. We can construct dual Plücker matrix [1] $\mathrm{L}_w^* = \boldsymbol{\pi}_1\boldsymbol{\pi}_2^{\mathrm{T}} - \boldsymbol{\pi}_2\boldsymbol{\pi}_1^{\mathrm{T}} \in \mathbb{R}^{4\times 4}$. Because dual Plücker matrix has the properties,

$$\mathrm{L}^* = \begin{bmatrix} [\mathbf{d}]_\times & \mathbf{m} \\ -\mathbf{m}^{\mathrm{T}} & 0 \end{bmatrix}, \quad (24)$$

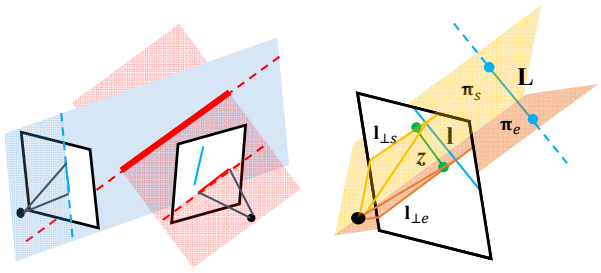
we can directly extract Plücker coordinates $(\mathbf{m}^{\mathrm{T}}, \mathbf{d}^{\mathrm{T}})^{\mathrm{T}}$.

However, Reconstruction using two frames has degeneracy when the measured line lies on epipolar plane that is discussed in [1]. Furthermore, it mis-creates 3D lines when a line is mismatched because Equation (23) and Equation (24) generates 3D line unconditionally whatever the matching is right as illustrated in Figure 5 (a).

We use n-views 3D line reconstruction to address these problems. To reconstruct 3D line by n-views, all planes $\mathbf{l}_{\mathrm{i}}^{\mathrm{T}}\mathrm{P}_{\mathrm{i}}, (i = 1 \cdots n)$ are stacked on matrix $\mathcal{W}$:

$$\mathcal{W} = \begin{bmatrix} \mathbf{l}_1^{\mathrm{T}}\mathrm{P}_1 \\ \mathbf{l}_2^{\mathrm{T}}\mathrm{P}_2 \\ \vdots \\ \mathbf{l}_n^{\mathrm{T}}\mathrm{P}_n \end{bmatrix}_{n\times 4}. \quad (25)$$

By singular value decomposition of $\mathcal{W}$ as $[\mathrm{S}, \mathrm{D}, \mathrm{V}] = \mathrm{SVD}(\mathcal{W})$, we get two dominant planes $\boldsymbol{\pi}_1$, $\boldsymbol{\pi}_2$ by taking two



(a) (b)

Figure 5: Illustrations for line reconstruction. (a) A case of line degeneracy that even false match generates 3D line. The blue line segment on right frame is right line pair, and the red line segment is incorrectly matched line. (b) Restoration of endpoints of 3D line **L**. The endpoints of 3D line can be found intersecting each plane $\boldsymbol{\pi}$ constructed by the perpendicular line $\mathbf{l}_{\perp}$ to projected line **l**, which passes each endpoint of detected line segment z.

columns of V corresponding to the two largest singular values. Then, 3D line can be reconstructed as same way by Equation (23) and Equation (24).

After reconstructing 3D line, it needs to recover two 3D endpoints for visualization. Because the measured endpoints projected from 3D endpoints of line are similar to endpoints of observed line segment, we use the way to use intersection plane suggested in [18] rather than directly back-projection of observed endpoints due to noise as shown in Figure 5 (b).

Given an observed line segment $\boldsymbol{z} = \{\mathbf{x}_s, \mathbf{x}_e\}$ and estimated line **l**, in the case of starting point, we compute closest points $\mathbf{x}_{\perp s}$ of $\mathbf{x}_s$ to **l** by calculating intersection point with a line $\mathbf{l}_{\perp s}$ perpendicular to **l** as:

$$x_{\perp s} = -\left(y_s - \frac{l_2}{l_1}x_s + \frac{l_3}{l_2}\right)\frac{l_1 l_2}{l_1^2 + l_2^2}, \tag{26}$$

$$y_{\perp s} = -\frac{l_1}{l_2}x_s - \frac{l_3}{l_2}. \tag{27}$$

Calculating $\mathbf{x}_{0s}$ intersecting x=0 with lying on $\mathbf{l}_{\perp s}$ as:

$$x_{0s} = 0, y_{0s} = y_s - \frac{l_2}{l_1}x_s. \tag{28}$$

We can compute 3D plane by

$$\boldsymbol{\pi}_s = \mathrm{P}^{\mathrm{T}}\mathbf{l}_{cs}, \text{where } \mathbf{l}_{cs} = \mathbf{x}_{\perp s} \times \mathbf{x}_{0s}. \tag{29}$$

Given Plücker line coordinates, $\mathbf{L} = (\mathbf{m}^{\mathrm{T}}, \mathbf{d}^{\mathrm{T}})^{\mathrm{T}}$, we can construct Plücker matrix L and 3D starting endpoint $\mathbf{D}_s$ can be recovered as:

$$\mathbf{D}_s = \mathrm{L}\boldsymbol{\pi}_s, \text{where } \mathrm{L} = \begin{bmatrix} [\mathbf{m}]_{\times} & \mathbf{d} \\ -\mathbf{d}^{\mathrm{T}} & 0 \end{bmatrix}, \tag{30}$$

and this process is done for ending point as well.

For n-view reconstruction, we select three-views(initial frame, middle frame, and current frame) for initialization. For local mapping, three closest key frames from current key frame are selected. A 3D line generated by three-views is rejected if the line does not satisfy i)Klein quadric constraints $\mathbf{m}^{\mathrm{T}}\mathbf{d}$ <0.01 or ii)perpendicular distance between projected line and endpoints of corresponding line segment is less than one in any view. If a line is failed to be generated, then it is tried to be generated by two-view reconstruction using the criterion of point feature reconstruction with regard to each endpoint suggested in [10] to prevent endpoints shifting. All the re-projection error for line is calculated by perpendicular distance that should be less than one as well.

4. Implementation details

We implement the proposed system with Intel Core i7-7700HQ (2.80GHz), 8GB memory and codes are written by C++. The proposed initialization and line-based SLAM are used for the proposed point-line SLAM system. This system is built on top of ORB-SLAM [10], and we implement line optimization based on g2o [31] pose-graph optimization framework. Therefore, the system architecture is same with ORB-SLAM except for i) line features are utilized simultaneously with ORB point features, and ii) initialization is conducted by the proposed matrix factorization. We use LSD line segment detector [32] to detect line segments, and LBD line binary descriptor [33] to descript the line segments as features. Initial camera poses to be used for matrix factorization are obtained by 8-points algorithm for rotation matrix and 2-points algorithm [27] for translation matrix. We only use robust frames for initialization by checking inliers by RANSAC scheme.

For line matching, we reject line-matching pairs if min line distance divided by max line distance are less than 0.8 to check distance similarity. In addition, if the difference of angle between line pairs are larger than angle between rotation axis of the corresponding two frames' rotation matrix, the line pairs are rejected.

5. Experiments

We compare the proposed system with other state-of-the-art systems in *TUM RGB-D Benchmark* [35]. In order to measure the precise effects with regard to the proposed initialization and line representation, we experiment each part separately.

For the comparison with other initialization, we compare the proposed initialization and conventional initialization. The conventional initialization uses a method suggested in ORB-SLAM [10], which selects a model either fundamental matrix or homography for pose estimation, and reconstructs landmarks using estimated poses. Both of systems are built on top of ORB-SLAM.

For the comparison of line representation, we call the

proposed system as RIPPL-SLAM (Robust Initialization and Plücker-based Point and Line SLAM). We compare RIPPL-SLAM with EPL-SLAM(Endpoints representation-based Point and Line SLAM) proposed in [12], and PPL-SLAM (Plücker-based Point and Line SLAM) proposed in [20]. EPL-SLAM uses endpoints representation for 3D line, and PPL-SLAM utilizes Plücker coordinates to represent 3D lines with different Jacobians of lines with the proposed system. EPL-SLAM and PPL-SLAM that are proposed on conventional initialization, and re-implemented. PPL-SLAM is revised as a monocular system because it is originally suggested for stereo system.

For the metrics used for the experiments, ATE (Absolute Trajectory Error) and RPE (Relative Pose Error) are used, which are provided from *TUM RGB-D Benchmark*. Because the map on a monocular SLAM is generated up to scale, we evaluate two trajectories, ground truth and estimated one, after aligning the scale. All the experiments are conducted without loop-closing to measure precise effects on different approaches for localization. Note that we test initialization using 2000 feature points per frame, which is basically suggested the number of features, and we do not report GSLAM because the KLT tracking is lost in most sequences.

The experimental result tested on different initialization is shown in Table 1. In the result, the proposed initialization is better than traditional initialization method. In addition, we emphasize that the proposed initialization generates a consistent and accurate map repeatedly. It is because the proposed method consumes multi-frames that reduces the influence of randomness. In contrast, the traditional initialization utilizes only two frames, thus the generated map is frequently influenced by random noises. Figure 6 shows an example of localization and mapping. Even though loop-closing module is de-activated, the SLAM that uses the proposed initialization constantly generates an accurate map met at the same locations in every trial. However, traditional initialization sometimes fails to meet same locations. Furthermore, Figure 7 shows that the proposed initialization generates an initial map which is converged faster than traditional initialization. Especially, the proposed method is also robust to the scenes including low-parallax cases, planar scene, or forward movement in which traditional methods have weakness due to ambiguity.

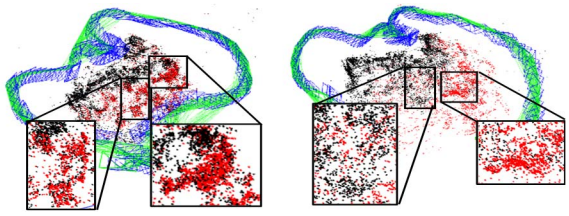
Figure 8 shows evaluation of the initialization methods w.r.t. the different number of features or frames. We operate SLAM using only first 100 frames to analyze only initial map. For this, we only utilize RPE metric, and we set error to one if initial map is not generated until 100 frames. Figure 8 (a) shows average RPE with converged frame number of TUM dataset. As the m increases, the convergence rate slows, but the accuracy increases. It is most accurate to use m=3. Nevertheless, regardless of m, the proposed method is better than conventional method.

Figure 8 (b) shows the performance according to the number of detected feature points. We utilize m =3 for the proposed method with and without the outlier rejection to analyze the effect of the accuracy of the essential matrix. In the conventional initialization, the fewer the features, the lower the performance due to strict criteria. In contrast, the proposed method is good even at lower features because this method has no strict criteria, and the factorization is tolerant to fewer features. Furthermore, the proposed method utilizing outlier rejection is better than not. This is because many inliers affect accurate estimation.

In the test for line representation, Table 2 shows the

Table 1: The experiment of different initialization method without loop-closing. This table shows localization accuracy on Absolute Trajectory RMSE [cm] and Relative Pose Error [cm], using 2000 feature points.

TUM RGB-D Sequence	*Proposed ini.*		*Conventional ini.*	
	ATE	RPE	ATE	RPE
f1_xyz	**1.0178**	**1.6275**	1.2514	1.7320
f2_xyz	**0.3062**	**1.5135**	0.3655	1.5413
f1_floor	3.4938	4.7756	**3.4003**	**4.7493**
f2_desk	**4.6464**	**6.7241**	4.6942	7.2342
f3_long_office	**2.9792**	**4.1529**	3.0770	4.1896
f3_nstr_tex_far	**3.6836**	**7.8440**	ambiguity	ambiguity
f3_sit_static	0.5934	0.9457	**0.3783**	**0.7105**
f3_str_tex_far	0.8701	1.5426	**0.7737**	**1.3942**



(a) (b)

Figure 6: Comparison of generated maps. (a) Using proposed initialization. (b) Using traditional initialization. Black dots indicate generated landmarks when the system is started, and red dots show generated landmarks when the position is revisited.

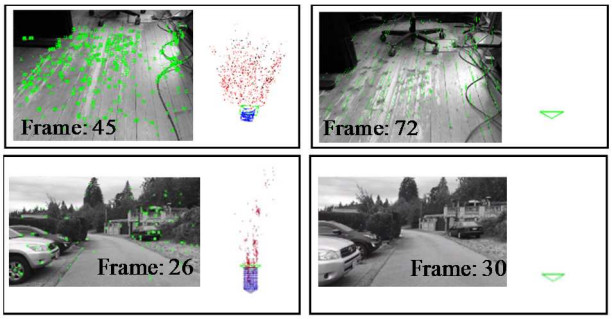
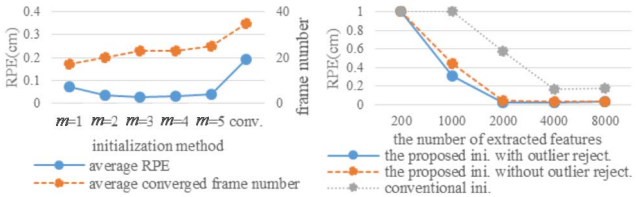


Figure 7: Results on different initialization. Left column denotes the use of proposed initialization, and right column indicates the failure cases of traditional initialization. The dataset of the first row is *f1_floor*, and the second row is another video taken by slow and forward motion for low-parallax case.



(a) (b)

Figure 8. Comparisons among different initialization methods. (a) comparisons of the average RPE and converged frame umber. (b) comparisons of the different number of extracted features.

results of comparisons among the proposed point-line SLAM system and the state-of-the-art point-line SLAM systems. The test is conducted by using only 500 feature points and 300 lines rather than using 2000 points per frame to reduce the effects of points and raise up the effects of lines. As Table 2 illustrates, the proposed method shows better performance. Especially, as shown in Figure 9, the proposed method performs precise localization even in a planar scene while other methods are failed. Furthermore, the reconstructed 3D lines are very clear and robust because those are generated by measuring multi-view consistency and robust criterion to prevent endpoints shifting. As shown in Figure 10 and Figure 11, the proposed method reconstructs very clear and robust shapes of 3D lines. In contrast, other methods generate a lot of inaccurate lines due to shifted endpoints or degeneracy of erroneously matched line pairs. The robust and clear 3D lines lead the localization of the system to be more accurate.

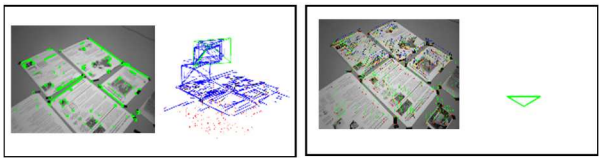
The execution time of the proposed system is similar to EPL-SLAM reported in [12]. However, local BA is faster than EPL-SLAM about 1.5 times because it optimizes only four parameters for lines while EPL-SLAM optimizes six parameters as using two endpoints. Therefore using Plücker coordinates is more memory efficient as well as faster. The geometric interpolation spends $m \times 1\text{ms}$ where m is subsequent frames used for matrix factorization, and matrix factorization is done at nearly 0.6ms. Thus, the proposed system can be executed as real-time system.

6. Conclusion

This paper presents an elaborate monocular indirect SLAM using proposed initialization and line features. The proposed initialization utilizes matrix factorization, which is applied to indirect SLAM by the proposed interpolation method with computational trick. In addition, this paper utilizes Plücker line coordinates and their orthonormal representation to calculate analytical derivations of lines. Experiments show that the proposed initialization generates an initial map in challenging scenes with fast and robust convergence. An accurate initial map also influences more accurate localization, and moreover, the proposed line representations improve the accuracy of localization with reduced computation and memory cost. We believe that the proposed initialization and localization can be used in featureless environments, especially in indoor scene. We will enhance outlier rejection and speed for more accurate and fast initialization and localization.

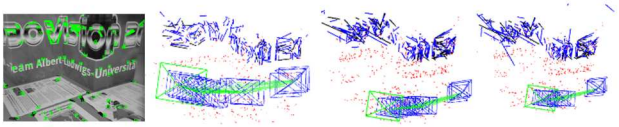
Table 2: The experiment of different line representation without loop-closing. This table shows localization accuracy on Absolute Trajectory RMSE [cm], using 500 feature points and 300 feature lines.

TUM RGB-D Sequence	RIPPL-SLAM (proposed)	EPL-SLAM	PPL-SLAM
f1_xyz	**0.9399**	1.1873	1.0816
f2_xyz	0.3907	0.3597	**0.3436**
f1_floor	3.4988	**3.3378**	4.7881
f2_desk	5.5371	**5.3653**	7.0389
f3_long_office	**1.5530**	1.8460	ambiguity
f3_nstr_tex_far	**1.5060**	ambiguity	ambiguity
f3_str_tex_far	**1.0059**	1.0294	1.0518



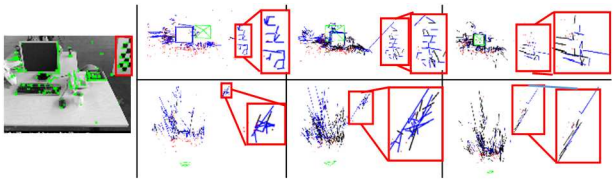
(a) (b)

Figure 9: Results on *f3_nstr_tex_far*. Image (a) denotes the proposed RIPPL-SLAM, and image (b) denotes EPL-SLAM and PPL-SLAM.



(a) An image. (b) RIPPL-SLAM (c) EPL-SLAM (d) PPL-SLAM

Figure 10: Clearness of reconstructed 3D lines on each method.



(a) An image (b) RIPPL-SLAM (c) EPL-SLAM (d) PPL-SLAM

Figure 11: Robustness of reconstructed 3D lines on each method. The area rounding red box in image (a) is used for analysis of the shapes reconstructed by 3D lines on each method in red boxes. The first row images are taken on front view, and the second row images are taken on top view.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2016R1D1A3B03934808).

References

[1] Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, 2nd ed. Cambridge University Press, 2004.

[2] Phillip H. Torr. Geometric motion segmentation and model selection. Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 356(1740): 1321-1340, 1998.

[3] Christopher H. Longuet-Higgins. The reconstruction of a plane surface from two perspective projections, Proceedings of the Royal Society of London. Series B. Biological Sciences, 227(1249): 399–410, 1986.

[4] Jakob Engel, et al. Lsd-slam: Large-scale direct monocular slam. In Proc. of ECCV,834-849, 2014.

[5] Jakob Engel, et al. Direct sparse odometry. IEEE transactions on pattern analysis and machine intelligence, 40(3): 611-625, 2018.

[6] Paul Bergmann, et al. Online photometric calibration of auto exposure video for realtime visual odometry and SLAM. IEEE Robotics and Automation Letters, 3(2): 627-634, 2018.

[7] Chengzhou Tang, et al. Gslam: Initialization-robust monocular visual slam via global structure-from-motion. 2017 International Conference on 3D Vision (3DV). IEEE, 155-164, 2017.

[8] Carlo Tomasi and T. Kanade. Detection and Tracking of Point Features. Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.

[9] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. Int. J. Comput. Vision, 56(3):221–255, 2004.

[10] Raul Mur-Artal, et al. Orb-slam: a versatile and accurate monocular slam system. IEEE Trans. Rob., 31(5):1147–1163, 2015.

[11] Raul Mur-Artal, and Jaun D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Transactions on Robotics, 33(5): 1255-1262, 2017.

[12] Albert Pumarola, et al. PL-SLAM: Real-time monocular visual SLAM with points and lines. 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 4503-4508, 2017.

[13] Georges Younes, et al,. Keyframe-based monocular SLAM: design, survey, and future directions. Robotics and Autonomous Systems, 98, 67-88, 2017.

[14] Georg Klein and David Murray, Parallel tracking and mapping for small AR workspaces," in IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR), Nara, Japan, November 2007,1-10, 2017.

[15] Christian Forster, et al. SVO: Fast semi-direct monocular visual odometry. In 2014 IEEE international conference on robotics and automation (ICRA) IEEE,15-22, 2014.

[16] Nianjuan Jiang, et al. A global linear method for camera pose registration. In Proc. of ICCV, 481-488, 2013.

[17] Zhaopeng Cui and Ping Tan. Global structure-from-motion by similarity averaging. In Proc. of ICCV, 864-872, 2015.

[18] Guoxuan Zhang, et al. Building a 3-d line-based map using stereo slam, IEEE Transactions on Robotics, 31(6): 1364–1377, 2015.

[19] Yan Lu and Dezhen Song. Robust rgb-d odometry using point and line features, in Proceedings of the IEEE International Conference on Computer Vision, 3934–3942, 2015.

[20] Xingxing Zuo, et al. Robust visual SLAM with point and line features. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 1775-1782, 2017.

[21] Thomas Lemaire, and Simon Lacroix. Monocular-vision based SLAM using line segments. In Proceedings 2007 IEEE International Conference on Robotics and Automation, 2791-2796, 2007.

[22] Joan Sola, et al. Undelayed initialization of line segments in monocular SLAM. In 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems 1553-1558, 2009.

[23] Dong Ruifang, et al. Line-based monocular graph SLAM. In 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), 494-500, 2017.

[24] Lilian Zhang and Reinhard Koch. Structure and motion from line correspondences: Representation, projection, initialization and sparse bundle adjustment, Journal of Visual Communication and Image Representation, 25(5): 904–915, 2014.

[25] Wojciech Chojnacki, and Michael J. Brooks. Revisiting Hartley's normalized eight-point algorithm. IEEE transactions on pattern analysis and machine intelligence, 25(9): 1172-1177, 2003.

[26] Luarent Kneip, et al. Finding the exact rotation between two images independently of the translation. In Proc. of ECCV, 696-709, 2012.

[27] Luarent Kneip and Rolend Y. Siegwart. Robust real-time visual odometry with a single camera and an imu. In Proc. BMVC, 16.1–16.11, 2011.

[28] Comelius Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, J. Res. Nat'l Bur. Std. 45, 255-282, 1950.

[29] Adrien Bartoli, and Peter Sturm. Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. Computer vision and image understanding, 100(3): 416-441, 2005.

[30] Hall. B. C. Lie Groups, Lie Algebras, and Representations: An Elementary Introduction, Graduate Texts in Mathematics, 222 (2nd ed.), Springer, ISBN 978-3319134666, 2015.

[31] Rainer Kuemmerle, et al. g2o: A general framework for graph optimization, in IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, May 2011, 3607–3613, 2011.

[32] Rafael Grompone Von Gioi, et al. LSD: a line segment detector. Image Processing On Line, 2: 35-55, 2012.

[33] Lilian Zhang, and Reinhard Koch. An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency. Journal of Visual Communication and Image Representation, 24(7): 794-805, 2013.

[34] Matthew. T. Mason. Mechanics of Robotic Manipulation. The MIT Press, 2001.

[35] Jurgen Sturm, et al. A benchmark for the evaluation of RGB-D SLAM systems. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE. 573-580, 2012.

[36] Amit Goldstein, and Raanan Fattal. Video stabilization using epipolar geometry. ACM Transactions on Graphics, 31(5): 126, 2012.