Visual-Inertial Odometry by Point and Line Features Under Filtration and Selection

Xitian Gao, Xiaojing He, Baoquan Li(✉), and Wuxi Shi

Tiangong University, Tianjin 300387, China
libq@tiangong.edu.cn

Abstract. For the visual-inertial odometry (VIO) system, the localization performance depends heavily on initialization for the camera module. Generally, line features are abundant in workspace, which contain rich geometrical information and can be captured steadily. In this paper, a novel initialization pattern is proposed for monocular visual-inertial localization, which utilizes both point and line features for camera pose estimation. Firstly, a filtration strategy is designed with respect to lengths of line features, aiming at reducing time cost and computational complexity of line feature extraction. As line features can express plane information well, reconstruction approaches are then employed for planar and nonplanar scenes by a selection criterion, so as to improve accuracy of pose estimation. At last, localization information is obtained by fusing point-line features with inertial measurement unit (IMU) measurements to increase robustness of the VIO system. Comparative experiments on public datasets are conducted to validate performance of the proposed method.

Keywords: Visual-inertial odometry · Initialization · Point-line features

1 Introduction

With continuous advancement of localization and navigation technologies, various robots can accomplish complex tasks independently in many scenarios [1]. Traditionally, only one sensor is used for collecting environment information in robotic platforms, and the most common sensor is the monocular camera. A vision sensor can acquire rich information from external environment, but frequency of data collection is relatively low with respect to quick movement. To improve localization accuracy, fusing multiple sensors for precise state estimation becomes one of the hottest topics in robotic fields [2]. One typical framework is to fuse a camera and an IMU to realize the visual-inertial odometry (VIO) [3].

This work was supported in part by the National Natural Science Foundation of China under Grant 61973234, in part by the Tianjin Natural Science Foundation under Grant 18JCZDJC96700 and Grant 20JCYBJC00180.

W. Fu et al. (Eds.): ICAUS 2022, LNEE 1010, pp. 2286–2300, 2023.
https://doi.org/10.1007/978-981-99-0479-2_213

Compared with cameras, IMUs can provide high frequency samplings, but accumulate errors as they work. Therefore, fusing a camera and an IMU can make full use of advantages of both sensors for accurate localization. In the initialization stage of VIO systems, most of existing methods only use point features for visual measurement processing, however, point extraction is challenging in textureless and illumination variation scenes. On the other hand, it brings merits to extract other types of features, such as line features, in environment for designing VIO strategies.

For VIO approaches, based on fusion pattern of visual and inertial measurements, they can be classified into loosely-coupled and tightly-coupled categories. For loosely-coupled approaches, visual and inertial measurements are used for pose estimation separately, and then results of the two estimators are fused to obtain the final localization information. Tightly-coupled approaches use raw data from the two sensors jointly for pose estimation, which can generally obtain more accurate and robust results compared to loosely-coupled ones [4]. On the other hand, according to fusing frameworks for visual and inertial measurements, VIO approaches can also be classified as filter-based [5] and optimization-based categories [6]. Filter-based approaches are designed with extended Kalman filters (EKF), which use IMU measurements for pose prediction and update the results by visual measurements [7]. Mourikis *et al.* propose the classic multi-state constraint Kalman filter method (MSCKF), where computation complexity is reduced by marginalizing landmark coordinates from the state vector but instant visual measurements are not fully used [8]. Li *et al.* further enhance the MSCKF algorithm with closed-form expression [9]. For the optimization-based approaches, camera and IMU samplings are optimized iteratively with preintegration techniques to improve localization performance [10]. Classic optimization-based approaches include the keyframe-based visual-inertial SLAM method (OKVIS) designed by Leutenegger *et al.* [11] and the versatile monocular visual-inertial state estimator (VINS-Mono) designed by Qin *et al.* [12].

Visual odometry is divided into categories as feature-based odometry, direct sparse odometry, and semi-direct odometry, according to whether it needs extract image features. It is noted that feature-based methods are able to match images with robustness and efficiency, and have good invariance to changes of viewpoints as well as illumination. Over the years, most of visual odometry methods extract only point features for pose estimation [8,13]. Representative pixels in images define the point features, and typical methods of extracting point features are SIFT, FAST, and ORB [14]. With executing matching operations after extracting point features, relative pose transformation between specific two frames is recovered with reconstruction algorithms, and for the sliding window pattern, triangulate and perspective-n-point (PnP) algorithms are used for reconstructing multiple frames. Klein *et al.* design the classic parallel tracking and mapping (PTAM) strategy by utilizing point features [15]. On this basis, Mur-Artal *et al.* propose the famous ORB-SLAM strategy by utilizing ORB features, which have the advantages of scale and rotation invariance [16]. The VINS-Mono method is designed by utilizing point features, with high-precision suitably for unmanned aerial vehicle (UAV) localization and mapping objectives, and this method has become a VINS benchmark [12].

In VIO systems, it is challenging to estimate pose information via point features in textureless and illumination varying environments. Line features are rich in workspaces generally and are as supplementary to point features in low-texture environments, and line detection is less sensitive to illumination variation and has more robust performance. Moreover, line segments provide more geometrical structure information for describing environments than point features. There are two kinds of representation for line segments: orthogonal coordinate representation and Plucker coordinate representation, where the orthogonal one uses three DoFs rotation and a scale factor for denotation, while the Plucker one is with over-parameterization to describe the four degree-of-freedoms (DoFs) spatial lines. As a result, it is reliable to make use of point and line features jointly for camera pose estimation, such as for visual odometry systems [17,18]. For visual-inertial odometry systems, point and line features can be fused with IMU measurements jointly. Kong *et al.* design a stereo visual-inertial system combining point and line features based on a filtering pattern [19]. Yu *et al.* parameterize line features for VIO systems to take the advantage of linearity properties [20]. Kottas *et al.* use only line features for pose estimation and provide observability analysis [21]. In [22], He *et al.* propose the PL-VIO strategy based on the optimization framework, which uses point and line features for tightly-coupled monocular VIO system.

In this paper, a novel visual-inertial odometry strategy is proposed by utilizing both point and line features for pose estimation, where visual perception is enhanced for the initialization stage of the camera module, so as to enhance performance of the whole VIO system. Firstly, a screening mechanism is designed for eliminating short line features, which can improve efficiency of line feature extraction and reduce time costs. Then, by taking it into account that line features have close relationship with plane information with respect to the workspace, the homography and fundamental matrix are calculated separately for plane and non-planar scenes, respectively. Accuracy of initial pose estimation is improved through the selection algorithm in the initialization stage. Lastly, the extracted line and point features are combined with IMU measurements for localization. Experimental results with respect to datasets are provided to verify performance of the proposed approach. Compared with existing methods that use point and line features for localization, main contributions of the paper are as follows: 1) complexity and time costs of line feature extraction are reduced by effectively filtrating short line segments, 2) robustness of pose estimation is enhanced by the selection algorithm for homography and fundamental matrix.

2 Problem Formulation

2.1 System Description

The framework of the proposed method is illustrated in Fig. 1. Since time cost is high for line feature extraction generally, computation complexity is reduced by filtrating short line features so as to ensure real-time performance for visual-inertial odometry systems. As line features can make good use of plane information

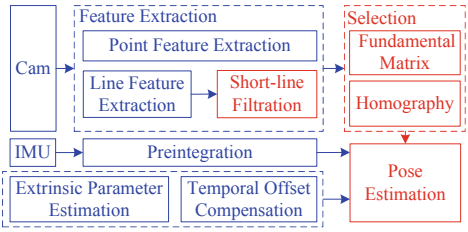

Fig. 1. Block diagram for visual-inertial odometry initialization with point and line features.

and point features can satisfy epipolar constraints, homography and fundamental matrix are used for line and point features, respectively. By judging plane and non-planar scenes, homography and fundamental matrix are utilized to obtain more accurate pose estimation results, respectively. Camera measurements are fused with the IMU module to estimate other states. As a result, by utilizing both point and line features to estimate pose information jointly, accurate localization is realized for VIO systems, even in some extreme scenes that are not rich with respect to point features.

2.2 Line Feature Representation

It is known that point features can be represented in three-dimensional (3D) coordinates directly with three DoFs. By contrast, spatial lines are often represented by two 3D endpoints with six DoFs, which results in over-parameterization in the sense that line segments actually have four DoFs. Plucker and orthogonal coordinate representations are two geometric descriptions for line segments.

Under the Plucker representation, a spatial line is expressed by two endpoints $\boldsymbol{S} = [x_1, y_1, z_1, \xi_1]^T$ and $\boldsymbol{E} = [x_2, y_2, z_2, \xi_2]^T$. So the line segment can be expressed under the camera coordinate system $\mathcal{F}^c$ as [23]

$$\mathcal{L} = \boldsymbol{S}\boldsymbol{E}^T - \boldsymbol{E}\boldsymbol{S}^T \tag{1}$$

where $\mathcal{L}$ is an antisymmetric matrix. Then, the Plucker matrix can be expressed as follows:

$$\mathcal{L} = \begin{bmatrix} [\boldsymbol{n}]_\times & \boldsymbol{d} \\ -\boldsymbol{d}^T & 0 \end{bmatrix} \tag{2}$$

where $\boldsymbol{n}(t)$ is the normal vector to the plane determined by this line segment with the camera optical center, $\boldsymbol{d}(t)$ is the line direction vector, and they are both expressed under $\mathcal{F}^c$ and have the constraint of $\boldsymbol{n}(t)^T\boldsymbol{d}(t) = 0$.

The $\mathcal{L}$ matrix in formula (2) has six DoFs, which is more than the four DoFs inherent in the line segment. Thus, this over-parameterization property brings additional constraints of two DoFs and increases computational complexity in the back-end optimization.

Unlike the Plucker representation, the orthogonal representation can avoid redundant constraint problems and speed up optimization processes. Plucker coordinates in (2) can be expressed under the orthogonal representation as $(U(t), W(t)) \in SO(3) \times SO(2)$, where $U(t)$ and $W(t)$ denote three and two dimensional rotation matrices, respectively. By applying the QR decomposition algorithm, $U(t)$ and $W(t)$ have the following form for describing line features under the orthogonal representation:

$$U = \left[\frac{\boldsymbol{n}}{\|\boldsymbol{n}\|}, \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|}, \frac{\boldsymbol{n} \times \boldsymbol{d}}{\|\boldsymbol{n} \times \boldsymbol{d}\|} \right], \quad W = \begin{bmatrix} w_1 & w_2 \\ -w_2 & w_1 \end{bmatrix} \tag{3}$$

where detailed meaning of $w_1(t)$ and $w_2(t)$ is shown in the following formula (5). Let $R(\boldsymbol{\varphi}) := U(t)$ and $R(\phi) := W(t)$, then we obtain

$$R(\boldsymbol{\varphi}) = [\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3] = \left[\frac{\boldsymbol{n}}{\|\boldsymbol{n}\|}, \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|}, \frac{\boldsymbol{n} \times \boldsymbol{d}}{\|\boldsymbol{n} \times \boldsymbol{d}\|} \right], \tag{4}$$

$$\begin{aligned} R(\phi) &= \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \\ &= \frac{1}{\sqrt{\|\boldsymbol{n}\|^2 + \|\boldsymbol{d}\|^2}} \begin{bmatrix} \|\boldsymbol{n}\| & -\|\boldsymbol{d}\| \\ \|\boldsymbol{d}\| & \|\boldsymbol{n}\| \end{bmatrix} \end{aligned} \tag{5}$$

where $\boldsymbol{\varphi}(t)$ and $\phi(t)$ represent a three-dimensional vector and a scale factor, respectively. Therefore, the orthogonal representation for a line feature has the form

$$\boldsymbol{o} = [\boldsymbol{\varphi}^T, \phi]^T. \tag{6}$$

As a result, transformation between the Plucker representation and the orthogonal representation is shown as follows:

$$\mathcal{L} = \left[w_1 \boldsymbol{u}_1{}^T, w_2 \boldsymbol{u}_2{}^T \right]. \tag{7}$$

The two representations are utilized in different stages of the visual-inertial SLAM system, respectively. The Plucker representation is utilized in front-end pose estimation while the orthogonal one is utilized in the back-end optimization, so as to improve calculation efficiency and avoid redundant constraints.

3 Line Feature Extraction and Matching

3.1 Line Feature Filtration

A large number of line features can be extracted from each image, and generally most of short-line features do not appear on two consecutive images [23]. In this sense, short-line features are not appropriate for feature matching and pose estimation. By contrast, long-line features have a higher probability of appearing in successive image frames and are easy for matching. Moreover, extraction of

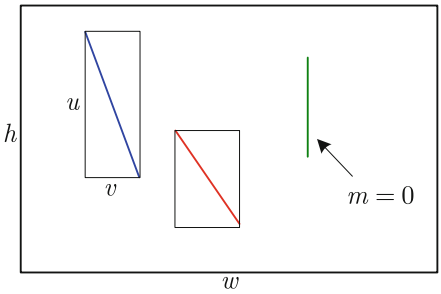

Fig. 2. Relationship between lengths of lines and their rectangle areas.

line features is relatively complicated and time-consuming. To deal with these problems, a filtration strategy needs to be designed with respect to lengths of line features.

Pixel coordinates of the start and the end points of a line segment are $(u_1(t), v_1(t))$ and $(u_2(t), v_2(t))$ in an image, respectively. Pixel distances are $u(t) = |u_1(t) - u_2(t)|$ and $v(t) = |v_1(t) - v_2(t)|$ in horizontal and vertical directions of the two endpoints, respectively, and the minimum value is denoted as $m(t) = \min\{u(t), v(t)\}$. Then, area relationship is evaluated between the rectangle of a line segment and the entire image to judge the length of the line feature.

1) If a line segment is vertical or horizontal ($m = 0$), as the vertical line segment in Fig. 2, then pixel length of the line segment is calculated through $l = \sqrt{u^2 + v^2}$ directly. Therefore, long line segments are retained for pose estimation and short ones are excluded directly, which are distinguished by the following criterion:

$$\begin{cases} \text{long} : l \geq \mu \min(h, w), \\ \text{short} : l < \mu \min(h, w) \end{cases} \tag{8}$$

   where h and w are height and width of the image, respectively, and μ is the ratio factor and is set to 0.125, the same as that in [23].
2) If the line segment is oblique ($m \neq 0$), the rectangle area of a line segment is expressed with $s = uv$, and another variable $t_s = s/m$ is defined for labeling longer line segments. As an example, rectangle areas of blue and red lines in Fig. 2 are the same, and the blue one is marked longer then the red one with bigger t_s.

Therefore, a line feature can be determined that whether it is a long segment or not by

$$\begin{cases} \text{long} : t_s > \eta \min(h, w), \\ \text{short} : t_s \leq \eta \min(h, w) \end{cases} \tag{9}$$

where η is a ratio factor. Trajectory estimation accuracy for the VIO system and extraction time of line features are contradictory with respect to the ratio factor

η. This factor is tried through practical manner for selecting a compromise value in the following experimental section. As a result, only long line features that satisfy requirements (8) and (9) are extracted appropriately for better feature matching and pose estimation.

3.2 Multiple View Geometry for Scenes

The fundamental matrix from multiple view geometry is suitable for accurate pose estimation in scenarios with rich point features, however, it degenerates and is sensitive to noise when the camera has pure rotation and observed feature points are coplanar. By contrast, line features can represent plane information well, and the homography can be used more suitably for pose estimation with respect to planar scenes. Therefore, both the two types of multiple view geometry are calculated in parallel to realize better visual initialization, and they are selected in planar and nonplanar scenes to estimate relative poses, respectively.

Relating to the fundamental matrix estimated by the eight-point algorithm, the essential matrix satisfies the epipolar constraint and has only five DoFs due to scale equivalence. The rotation matrix $R(t)$ and the translation vector $\boldsymbol{T}(t)$ are calculated through singular value decomposition (SVD) algorithms with depth evaluation. On the other hand, the homography is used to describe the mapping relationship with respect to plane scenes, such as grounds and walls. Similar to the essential matrix, pose information can be calculated from the homography by utilizing direct linear transform (DLT) algorithms.

The fundamental matrix and the homography are selected when point and line features are relatively richer, respectively. On the basis of the selection strategy in [16], scores $S_F(t)$ and $S_H(t)$, with unified form $S_M(t)$, are calculated separately for selecting the fundamental matrix and the homography in each iteration:

$$S_M = \sum_i \left(\rho_M(d_{cr}^2(p_c^i, p_r^i, M)) + \rho_M(d_{rc}^2(p_c^i, p_r^i, M))\right) \tag{10}$$

where

$$\rho_M(d_{cr}^2) = \begin{cases} \Gamma_M - d_{cr}^2, & d_{cr}^2 < T_M, \\ 0, & d_{cr}^2 \geq T_M \end{cases} \tag{11}$$

and $\rho_M(d_{rc}^2)$ is calculated with the same formula. In (10) and (11), symbol M denotes the selected fundamental matrix and homography, d_{cr}^2 and d_{rc}^2 are mutual reprojection errors from one frame to the other, T_M is the outlier rejection threshold with $T_F = 3.84$ for the fundamental matrix and $T_H = 5.99$ for the homography the same as those in [16], and Γ_M is defined similarly to T_M.

In special cases, such as parallax is low with approximate plane scenes, transformation relationship between two frames is ambiguous expressed by either the fundamental matrix or the homography. To solve this problem, the following criterion is utilized with respect to S_H and S_F:

$$R_H = \frac{S_H}{S_H + S_F}. \tag{12}$$

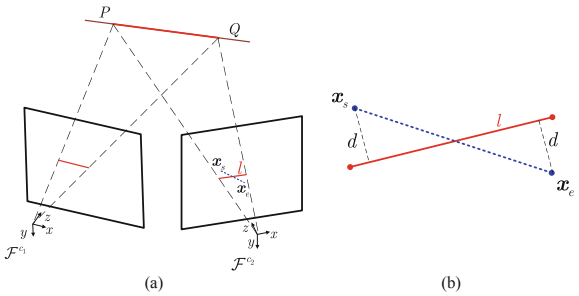

Fig. 3. The diagram for the reprojection error of a line feature.

with (12), the homography is selected when $R_H > 0.45$, otherwise the fundamental matrix is used. In summary, the above processes are designed to obtain better results of feature matching and pose estimation with point and line features.

3.3 Reprojection Errors of Line Features

The projection model of line segments with respect to the camera plane is

$$\boldsymbol{l} = K_l \boldsymbol{n} = \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ -f_y c_x & f_x c_y & f_x f_y \end{bmatrix} \boldsymbol{n} \tag{13}$$

where $\boldsymbol{l} = [l_1, l_2, l_3]^T$ is the normal vector of the plane that is defined by the line and the camera optical center, K_l is the transformation matrix for line segment projection, and f_x, f_y, c_x, c_y are camera intrinsic parameters.

The reprojection error of a line feature is illustrated in Fig. 3, and there are two camera keyframes $\mathcal{F}^{c_1}$ and $\mathcal{F}^{c_2}$ taken as an example with respect to the world coordinate system $\mathcal{F}^w$. A line feature is represented by PQ, which projects on the two image planes according to the projection model (13). The line segment on the $\mathcal{F}^{c_1}$ image plane is detected by line segment detector (LSD) algorithms, and its reprojection line segment is calculated and shown as the blue one on $\mathcal{F}^{c_2}$ with $\boldsymbol{x}_s$ and $\boldsymbol{x}_e$ being the start and the end points, respectively. The reprojected line segment is matched with the detected one on $\mathcal{F}^{c_2}$, and the reprojection error exists as shown in Fig. 3(b).

The line feature can be converted from the world coordinate system $\mathcal{F}^w$ to the camera coordinate system $\mathcal{F}^c$ by the following relationship:

$$\mathcal{L}_c = {}^c_w\mathcal{T}\mathcal{L}_w, \quad {}^c_w\mathcal{T} = \begin{bmatrix} {}^c_wR & [{}^c_w\boldsymbol{T}]_\times {}^c_wR \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} \tag{14}$$

where ${}^w_cR(t)$ and ${}^w_c\boldsymbol{T}(t)$ are the rotation matrix and the translation vector of $\mathcal{F}^c$ under $\mathcal{F}^w$, respectively.

The reprojection error model of a line feature with respect to the image plane can be obtained as

$$\boldsymbol{e}_l = \left[\frac{\boldsymbol{x}_s^T \boldsymbol{l}}{\sqrt{l_1^2 + l_2^2}}, \frac{\boldsymbol{x}_s^T \boldsymbol{l}}{\sqrt{l_1^2 + l_2^2}} \right]^T \tag{15}$$

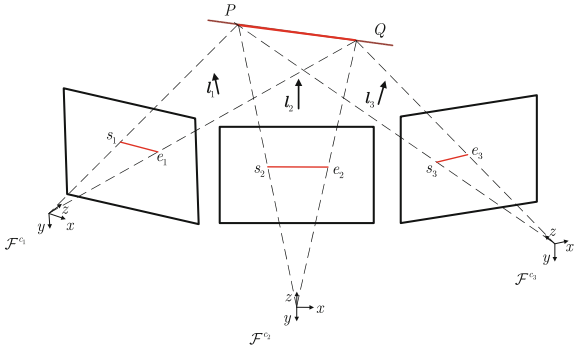

Fig. 4. The diagram of linear feature based pose estimation.

where $\boldsymbol{l}(t)$ is projected to the camera plane by (13), $l_1(t)$ and $l_2(t)$ are entities of $\boldsymbol{l}(t)$. The norm of $\boldsymbol{e}_l(t)$ is just the length of $d(t)$, as shown in Fig. 3(b).

4 Fusion of Line Features and the IMU

4.1 Pose Estimation with Line Features

Considering three successive camera keyframes $\mathcal{F}^{c_1}$, $\mathcal{F}^{c_2}$, and $\mathcal{F}^{c_3}$, a line segment PQ is observed on these images, as illustrated in Fig. 4. Line PQ projects on the three image planes with the features denoted as s_ie_i, $i = \{1, 2, 3\}$. Since keyframes are successive with short time, velocities among these three camera keyframes are assumed to be constant. Here the keyframe $\mathcal{F}^{c_2}$ is set as the reference, and then its rotation matrix denoted as R_2 is set equal to 3×3 identity matrix I [18]. Rotation matrices of $\mathcal{F}^{c_2}$ to $\mathcal{F}^{c_1}$ and $\mathcal{F}^{c_2}$ to $\mathcal{F}^{c_3}$ can be described as $R_{c_2}^{c_1} = R^T$ and $R_{c_2}^{c_3} = R$, respectively, where the rotation matrix R is expressed approximately as follows [18]:

$$R = \begin{bmatrix} 1 & -r_3 & r_2 \\ r_3 & 1 & -r_1 \\ -r_2 & r_1 & 1 \end{bmatrix}. \tag{16}$$

Camera optical center O_i and each projection line segments s_ie_i define a plane, and its normal vector is denoted as $\boldsymbol{l}_i$, $i = \{1, 2, 3\}$. The constraint for vectors $\boldsymbol{l}_1$, $\boldsymbol{l}_2$, and $\boldsymbol{l}_3$ is shown by the following equation:

$$\boldsymbol{l}_2{}^T((R^T\boldsymbol{l}_1) \times (R\boldsymbol{l}_3)) = 0. \tag{17}$$

Consequently, three quadratic equations are established by the above formula. Moreover, camera poses can be calculated for the initialization stage from (17), which is solved by employing polynomial solving algorithms in [24] with at least five matched line features.

4.2 IMU Fusion

Visual information becomes more sufficient by extracting both point and line features jointly. Thus, the vision module can estimate poses accurately in different scenes and reduce risk of tracking failure in lack of point features. Furthermore, after the visual module provides pose relationship between two camera keyframes, IMU measurements are aligned and fused to obtain other states. At last, the above algorithms are integrated into typical SLAM systems such as VINS-Mono, and localization performance is improved for the whole VIO system.

5 Experimental Results

In this section, experiments are conducted to verify performance of the proposed VIO method. The classic EuRoC datasets [25] are used with comparative tests to evaluate the accuracy of the proposed method.

5.1 Extraction of Long Line Features

The filtration strategy for long line features in Sect. 3. A is verified in this part, and the ratio factor η is obtained through tradeoff in experiments. Taking MH_05_difficult dataset as an example, feature maps of the workspace are shown in Fig. 5, where each subgraph corresponds to different ratio factors. The number of extracted line features becomes smaller as the ratio factor is adjusted larger, implying that lengths of extracted line features should be longer. On the other hand, with very few line features, it is difficult to combine line features with point features for robust pose estimation. Therefore, the ratio factor for line feature lengths can be turned to affect extraction time, moreover, pose estimation accuracy should also be taken into account when tuning this factor.

Figure 6 provides effect of different ratio factors by the filtrate strategy in the MH_05_difficult experiment. The top subgraph shows relationship between the scale factor and line feature extraction time, and the bottom one shows relationship between scale factor and root-mean-square-error (RMSE) of the estimated trajectory. It is seen that extraction time of line features decreases obviously when the scale factor increases, and RMSE of the estimated trajectory is preferable with a compromise value of the scale factor. As a result, the scale factor is selected as 0.2 by taking into account both extraction time and trajectory RMSE.

5.2 Tests for Localization

The EuRoC datasets provide 20 frames per second images, synchronized IMU measurements, and ground truth for states of unmanned aerial vehicles. The left camera from stereo vision is only used in this part as the monocular camera. All experiments are performed on a laptop computer (Quadcore Intel i7-7700H,

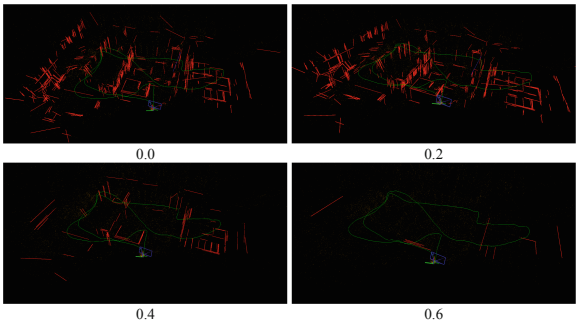

Fig. 5. Line feature maps corresponding to different ratio factors η in the MH_05_difficult experiment.

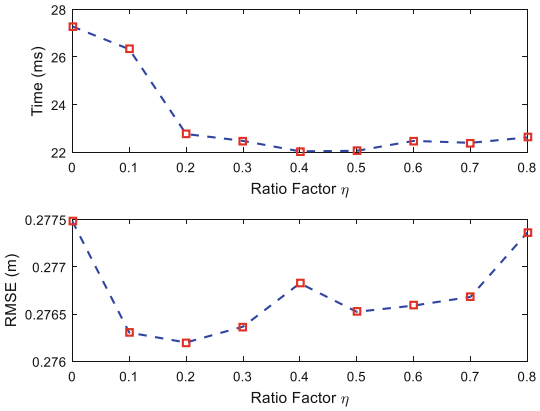

Fig. 6. Line feature extraction time (top) and trajectory RMSE (bottom) with respect to different ratio factors η in the MH_05_difficult experiment.

CPU@2.8 GHZ, 8 GB RAM) with Ubuntu 16.04/ROS Kinetic. We accomplish the proposed strategy on the basis of the PL-VINS framework [23], and provide our detailed implementation on the website[1]. To form the monocular visual-inertial SLAM system with respect to point-line features, the implementation includes real-time line feature extraction, loop detection, and relocalization.

Experimental results of the proposed method with relevant datasets are provided in Table 1, and localization errors are evaluated by RMSE from the evo package (https://github.com/MichaelGrupp/evo). Moreover, the PL-VINS method [23] is conducted as a comparison, and relevant results are also provided in Table 1. It is seen that localization accuracy of the proposed method is superior over that of the comparative method.

[1] https://github.com/TGUMobileVision/VIOPointLine.

Table 1. RMSE of the proposed method and PL-VINS.

Sequence	Proposed	PL-VINS
MH_01_easy	**0.152**	0.157
MH_02_easy	0.171	**0.169**
MH_03_medium	0.277	0.277
MH_04_difficult	**0.299**	0.303
MH_05_difficult	**0.276**	0.281
V1_02_medium	**0.122**	0.123
V1_03_difficult	**0.177**	0.181

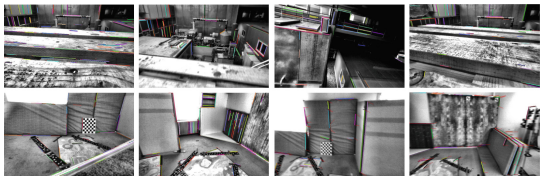

Fig. 7. Process images by the proposed method in MH_05_difficult (top) and V1_03_difficult (bottom) experiments.

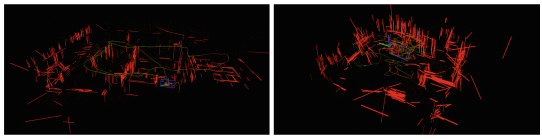

Fig. 8. Extracted point-line features and estimated trajectories by the proposed method in MH_05_difficult (left) and V1_03_difficult (right) experiments.

Process images of MH_05_difficult and V1_03_difficult experiments are shown in Fig. 7. It can be seen that abundant line features are detected in scenes with less texture and small parallax. Figure 8 provides both trajectories and feature maps for these experiments, where red and green lines denote ground truth and actual trajectories, respectively.

Moreover, trajectory errors are marked with graduated colors to display results intuitively, as shown in Fig. 9. Left and right subgraphs represent results of the proposed method and [23], respectively. It is seen that trajectory errors of the proposed method are less than that of PL-VINS, obviously when the camera rotates rapidly.

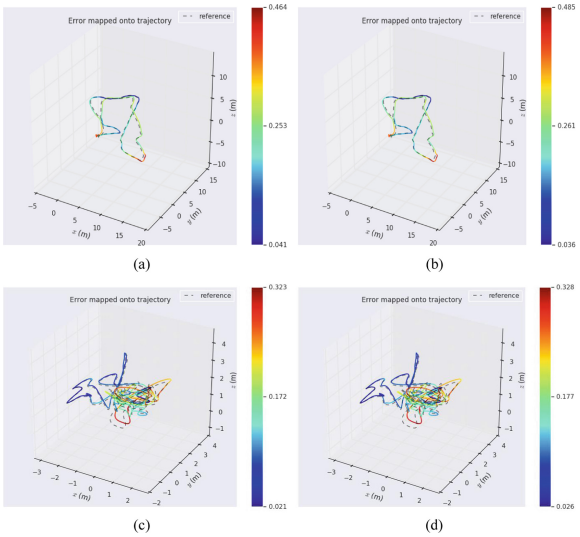

Fig. 9. Trajectory error comparison in MH_05_difficult (top) and V1_03_difficult (bottom) experiments (left subgraphs: the proposed method, right subgraphs: PL-VINS).

6 Conclusions

A novel visual-inertial odometry is proposed by utilizing point and line features jointly for pose estimation. For line feature extraction, to deal with the problem of high computational complexity, a filtration strategy is designed with respect to lengths of line features, so as to exclude short line features and speed up line feature extraction time to obtain real-time performance. Since point features satisfy the epipolar constraint and line features well express plane information, fundamental matrix and homography are set to be computed concurrently for non-planar and planar scenes, respectively. Then, a selection strategy is conducted for the two multiple view geometry results to improve localization accuracy. Pose information obtained by the camera module is finally fused with measurements of the IMU module, and the scale factor and other states are calculated to obtain more robust initialization information for the visual inertial odometry.

References

1. Qin, T., Shen, S.: Robust initialization of monocular visual-inertial estimation on aerial robots. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4225–4232 (2017)
2. Huang, G.: Visual-inertial navigation: a concise review. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 9572–9582 (2019)
3. Mur-Artal, R., Tardos, J.D.: Visual-inertial monocular SLAM with map reuse. IEEE Robot. Autom. Lett. **2**(2), 796–803 (2017)

4. Martinelli, A.: Closed-form solution of visual-inertial structure from motion. Int. J. Comput. Vis. **106**(2), 138–152 (2014)
5. Liu, Y., et al.: Stereo visual-inertial odometry with multiple Kalman filters ensemble. IEEE Trans. Ind. Electron. **63**(10), 6205–6216 (2016)
6. Shen, S., Michael, N., Kumar, V.: Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 5303–5310 (2015)
7. Heo, S., Jung, J.H., Park, C.G.: Consistent EKF-based visual-inertial navigation using points and lines. IEEE Sens. J. **18**(18), 7638–7649 (2018)
8. Mourikis, A.I., Roumeliotis, S.I.: A multi-state constraint Kalman filter for vision-aided inertial navigation. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 3565–3572 (2007)
9. Li, M., Mourikis, A.I.: High-precision, consistent EKF-based visual-inertial odometry. Int. J. Robot. Res. **32**(6), 690–711 (2013)
10. Forster, C., Carlone, L., Dellaert, F., Scaramuzza, D.: On-manifold preintegration for real-time visual-inertial odometry. IEEE Trans. Robot. **33**(1), 1–21 (2017)
11. Leutenegger, S., Furgale, P., Rabaud, V., Chli, M., Konolige, K., Siegwart, R.: Keyframe-based visual-inertial SLAM using nonlinear optimization. Int. J. Robot. Res. **34**(3), 314–334 (2015)
12. Qin, T., Li, P., Shen, S.: VINS-Mono: a robust and versatile monocular visual-inertial state estimator. IEEE Trans. Robot. **34**(4), 1004–1020 (2018)
13. Gao, X., Li, B., Shi, W., Yan, F.: Visual-inertial odometry system with simultaneous extrinsic parameters optimization. In: Proceedings of IEEE International Conference on Advanced Intelligent Mechatronics, pp. 1977–1982 (2020)
14. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: Proceedings of IEEE International Conference on Computer Vision, pp. 2564–2571 (2012)
15. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: Proceedings of IEEE International Symposium on Mixed and Augmented Reality, pp. 225–234 (2008)
16. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans. Robot. **31**(5), 1147–1163 (2017)
17. Gomez-Ojeda, R., Moreno, F.A., Scaramuzza, D., Gonzalez-Jimenez, J.: PL-SLAM: a stereo SLAM system through the combination of points and line segments. IEEE Trans. Robot. **35**(3), 734–746 (2017)
18. Pumarola, A., Vakhitov, A., Agudo, A., Sanfeliu, A., Moreno-Noguer, F.: PL-SLAM: real-time monocular visual SLAM with points and lines. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 4503–4508 (2017)
19. Kong, X., Wu, W., Zhang, L., Wang, Y.: Tightly-coupled stereo visual-inertial navigation using point and line features. Sensors **15**(7), 12816–12833 (2015)
20. Yu, H., Mourikis, A.I.: Vision-aided inertial navigation with line features and a rolling-shutter camera. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 892–899 (2015)
21. Kottas, D.G., Roumeliotis, S.I.: Efficient and consistent vision-aided inertial navigation using line observations. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 1540–1547 (2013)
22. He, Y., Zhao, J., Guo, Y., He, W., Yuan, K.: PL-VIO: tightly-coupled monocular visual-inertial odometry using point and line features. Sensors **18**(4), 1159 (2018)
23. Fu, Q., et al.: PL-VINS: real-time monocular visual-inertial SLAM with point and line features arXiv:2009.07462 (2020)

24. Kukelova, Z., Bujnak, M., Pajdla, T.: Polynomial eigenvalue solutions to minimal problems in computer vision. IEEE Trans. Pattern Anal. Mach. Intell. **34**(7), 1381–1393 (2012)
25. Burri, M., et al.: The EuRoC micro aerial vehicle datasets. Int. J. Robot. Res. **35**(10), 1157–1163 (2016)