

Highly Efficient Line Segment Tracking with an IMU-KLT Prediction and a Convex Geometric Distance Minimization

Hao Wei^{1,2}, Fulin Tang², Chaofan Zhang³ and Yihong Wu^{2,1*}

Abstract—Line segment features become popular in SLAM community. Usually, line-based SLAM systems utilize local appearance descriptors for line segment tracking. However, traditional descriptor-based line segment tracking algorithms suffer from the problem that accuracy and speed cannot be possessed simultaneously, which affects the performance of line-based SLAM systems negatively. We propose a novel line segment tracking method with an IMU-KLT line segment prediction and a convex geometric distance minimization to boost line segment tracking performance in both accuracy and speed. Particularly, the proposed convex geometric distance minimization uses a ℓ^1 -norm model to minimize geometric constraints between predicted line segments and extracted line segments efficiently. Furthermore, the line segment tracking is embedded into a VIO system and we adapt it to obtain more reliable point tracking. Experimental results on public datasets show that the proposed line segment tracking method achieves much higher accuracy and much less time cost than state-of-the-art level, where not only the number of correct matches increases but also the inlier ratios are increased by at least 35.1% along with a 3 times faster speed. Besides, the VIO system combining the proposed line segment tracking is improved in terms of accuracy.

I. INTRODUCTION

Nowadays, visual simultaneous localization and mapping (V-SLAM) or visual-inertial odometry (VIO) are popular due to their wide applications, such as autonomous driving, virtual reality, and robot navigation. Most V-SLAM/VIO systems [1]–[3] are based on point features. However, reliable point features are quite few in weakly textured scenes, while line features can be extracted, particularly in many human made scenes like office buildings or urban cityscapes. Moreover, line features can provide more meaningful structure information about surrounding environments [4]. Therefore, line segment features have been used in V-SLAM/VIO systems recently, such as [5]–[8].

However, there are still some challenges in line-based V-SLAM/VIO systems. Firstly, as described in [9], textures around line segments are approximate, which brings about difficulties for accurately tracking line segments by using local line descriptors. Secondly, tracking line segments based

*This work was supported by National Natural Science Foundation of China under Grant Nos. 61836015, 62002359 and supported by the Beijing Advanced Discipline Fund under Grant No. 115200S001. The corresponding authors are Fulin Tang and Yihong Wu. E-mail: fulin.tang@nlpr.ia.ac.cn and yhwu@nlpr.ia.ac.cn.

¹School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China.

²National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

³Anhui Institute of Optics and Fine Mechanics, Hefei Institutes of Physical Science, Chinese Academy of Sciences.

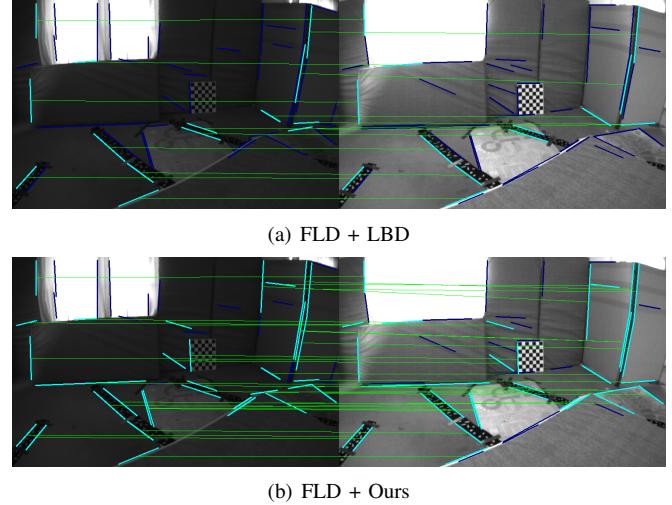


Fig. 1. Line segment tracking results (remove outliers) of LBD [13] and our method under challenging illumination changes scene. The image pairs are from the EuRoC dataset [31] and we use FLD [28] to detect line segments. Cyan lines show corresponding line segments, and blue lines show no corresponding line segments. In (a), there are 14 correspondences. In (b), there are 33 correspondences.

on local descriptors is extremely time-consuming. Because of these problems, line segment features have not been utilized widely, which motivates us to design a both accurate and fast line segment tracking algorithm.

We propose an accurate and simultaneously fast line segment tracking method, which does not rely on descriptors. We formulate the line segment tracking problem as a sparse and convex ℓ^1 -norm minimization according to the geometric constraints between predicted line segments and extracted line segments, which contain angle, point-wise distance and midpoint distance information. The predicted line segments are obtained by an IMU-aided KLT algorithm. Since our method utilizes strict geometric constraints, it achieves better performance compared with the descriptor-based method, LBD [13], as shown in Fig. 1. To further validate the proposed line tracking algorithm, we fuse it into an existing point-based VIO system, OpenVINS [1], where a method is proposed for converting line correspondences to point correspondences. The proposed point tracking method is based on epipolar and image appearance constraints.

The main contributions are summarized as follows.

- 1) An IMU-KLT line prediction is given, where we combine point prediction and line fitting. The result is more reliable.
- 2) An efficient geometric distance is given to measure

- the distance between two line segments, where we use angle, point-wise distance and midpoint distance information.
- 3) A line segment tracking algorithm fusing the above 1) and 2) is proposed, where a ℓ^1 -norm convex optimization to minimize the proposed distance between the line prediction and the extracted line segments is performed. As we can see, line descriptors are not used. The IMU-KLT line prediction and the geometric distance minimization are very reliable. Therefore, the proposed line segment tracking is both more accurate and faster relative to state-of-the-art level.
 - 4) The proposed line segment tracking is embedded into a VIO system by using epipolar and image appearance constraints, where we obtain very reliable point tracking.

- 5) Extensive experimental results validate that the proposed line segment tracking is highly efficient. Not only the matched inlier ratios are increased by at least 35.1% and up to 57.3% relative to state-of-the-art level but also the number of correct matches increases. The speed is also 3 times faster than them. Moreover, the VIO system with the proposed line segment tracking is improved in terms of accuracy.

This paper is organized as follows. We discuss the related work in Section II. The details of the proposed line segment tracking algorithm and the point tracking method are presented in Section III and Section IV, respectively. In Section V, we provide experiments. Conclusions and future work are given in Section VI.

II. RELATED WORK

A. Line Segment Tracking Algorithm

Line segment tracking plays an important role in line-based SLAM systems. There are different schemes for line segment tracking, which can be broadly categorized into geometry-based tracking algorithms, local descriptor-based algorithms, and deep learning-based algorithms. For geometry-based tracking algorithms, Fan et al. [9] encoded local geometric information between a line and its neighboring points as line-point invariants, and then the matched points are used to support line matching via the line-point invariant. Ruben et al. [11] proposed a purely geometrical line tracking approach for high dynamic range (HDR) environments. For local appearance descriptor-based algorithms, Wang et al. [12] presented a line descriptor called mean-standard deviation line descriptor (MSLD) for robust line matching without resorting to any prior knowledge. Line band descriptor (LBD) [13] is a line descriptor designed to characterize the local appearance, which is the most popular method in line-based SLAM systems. For deep learning-based algorithms, DLD [14] used ResNet [15] to learn an appearance-based line descriptor. Besides, Ma et al. [16] applied a graph convolution network to match line segments.

In consideration of accuracy and efficiency, we adopt a geometry-based tracking algorithm to design our method.

Different from [11], we utilize different geometric constraints, which contain angle, point-wise distance and midpoint distance between predicted line segments and extracted line segments. More importantly, we adopt an IMU-KLT method to predict line segments, which can offer a good initial value of the tracking step. Compared with the method [11], LBD [13] is more popular and widely used in SLAM systems. Moreover, [11] is not open-source. Therefore, we compare the proposed method with LBD in experiments.

B. Line-based SLAM

During the past decades, many successful SLAM/VIO systems have been proposed, such as ORB-SLAM [3], SVO [17], DSO [2], VINS-Mono [18], and OpenVINS [1]. Based on these work, some researchers try to improve the robustness of SLAM systems by combining different geometric features, such as edges [19], line segments [5], [6], [20] and planes [21].

Line segments are less sensitive to illumination changes and can be detected easily in man-made environments. Therefore, line segment features have been used in recent years. For example, in [5], [22], [23], line features are integrated into the ORB-SLAM system [3], and in [6], [24], line segment features are integrated into DSO [2]. In these papers, line segments are used to improve robustness and accuracy of the systems. Recently, line segments are used in a lidar-monocular visual odometry [7]. Besides, Zhou et al. [25], [26] incorporated line segments with prior orientation called structural lines as features.

Most existing line-based systems utilize LBD [13] descriptors for line segment tracking. Descriptor-based line tracking methods usually behave well in general scenes but with high computational burdens. Therefore, we propose a line segment tracking algorithm without using descriptors for low time-consumption. Simultaneously in order to higher accuracy, we propose an IMU-KLT line prediction and a convex geometric distance optimization. To apply the proposed algorithm into an existing point-based VIO system [1], we also propose a novel technique to convert the line segment correspondences to point correspondences. Then many well-established point SLAM or VIO systems can combine the proposed line segment tracking.

III. LINE SEGMENT TRACKING

A. Predicting

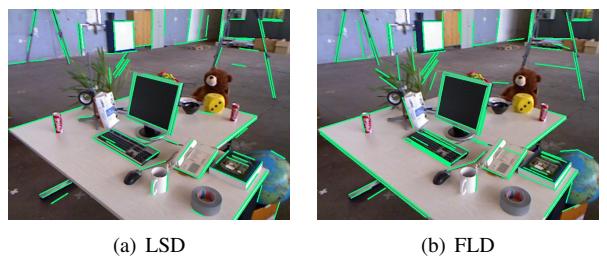


Fig. 2. Line segment extraction results of LSD [27] and FLD [28].

We extract line segments for each image first. At present, there are two popular line segment detectors, LSD [27] and FLD [28]. Although FLD [28] has a slightly inferior performance compared with LSD [27], they both satisfy requirements of SLAM systems. The line segment extraction results are shown in Fig. 2. What's more, FLD [28] is more efficient than LSD [27]. Thus, we select FLD [28] to extract line segments.

For each extracted line segment, we sample c points. If we detect m line segments in the current image frame, there are $m \cdot c$ points totally. Then, we use an IMU-aided KLT method to estimate their corresponding positions in the next image frame.

For two consecutive images I_1 and I_2 , \mathbf{R} and \mathbf{t} describe relative rotation and translation between them, and \mathbf{K} describes the camera intrinsic parameter matrix. Let \mathbf{u}_1 and \mathbf{u}_2 be a pair of corresponding pixel coordinates from I_1 and I_2 respectively. We transform them by \mathbf{K}^{-1} to obtain normalized coordinates, denoted as \mathbf{p}_1 and \mathbf{p}_2 . Moreover, we denote their depths as λ_1 and λ_2 . According to the transformation between I_1 and I_2 , we obtain:

$$\lambda_2 \mathbf{p}_2 = \lambda_1 \mathbf{R} \mathbf{p}_1 + \mathbf{t}. \quad (1)$$

Further, we have:

$$\lambda_2 \mathbf{K}^{-1} \mathbf{u}_2 = \lambda_1 \mathbf{R} \mathbf{K}^{-1} \mathbf{u}_1 + \mathbf{t}. \quad (2)$$

Usually, for two consecutive images, the translation vector \mathbf{t} is small which can be ignored. Also, we consider λ_1 is equal to λ_2 approximately. Therefore, (2) can be simplified as:

$$\mathbf{u}_2 = \mathbf{R} \mathbf{K}^{-1} \mathbf{u}_1, \quad (3)$$

where the rotation matrix \mathbf{R} is integrated from IMU sequences between I_1 and I_2 . Based on (3), for a feature point \mathbf{u}_1 in I_1 , we can offer a predicted position \mathbf{u}'_1 in I_2 . After obtaining \mathbf{u}'_1 , we use the KLT sparse optical flow to track \mathbf{u}'_1 in I_2 . Compared with the original KLT algorithm, our method uses (3) to give a more reliable prediction, which improves accuracy of point tracking in fast motion scenes.

For points belonging to a line segment \mathbf{l}_k in I_1 , their corresponding positions in I_2 are estimated by the above IMU-aided KLT algorithm, and then we remove outliers as follows:

- 1) Feature points tracked successfully by the above IMU-aided KLT algorithm and satisfying epipolar constraint are selected.
- 2) Feature points located or adjacent to image edges are selected.
- 3) We fit the points in the above 1) and 2) to a line by using a least-square method and then remove points that are far from the fitted result.

After the above three steps, we obtain the reliable tracked points in I_2 . Next, these points are used to fit a line segment again with a least-square method. For other line segments in I_1 , we repeat the above steps to get their predicted line segments.

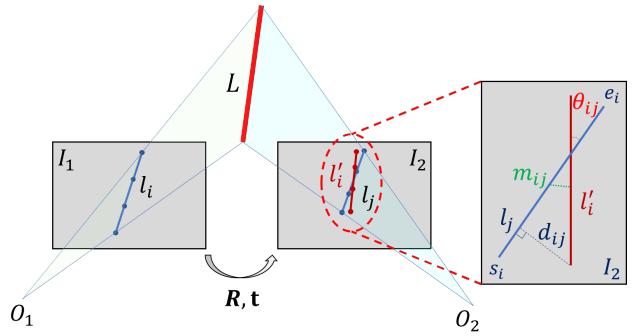


Fig. 3. Illustration of the proposed line segment tracking algorithm. For a line segment \mathbf{l}_i in I_1 , its predicted line segment in I_2 is \mathbf{l}'_i , and the correct correspondence of \mathbf{l}_i is \mathbf{l}_j .

B. Tracking

After getting the predicted line segments, we further find more accurate line segment correspondences from some candidates. The candidates are the extracted line segments in I_2 located in an area. The area in I_2 is the same as a neighbor of the tracked line segment in I_1 . We formulate the tracking process as minimizing the following three geometric constraints between the predicted line segments and the candidates.

Let \mathbf{s}_i and \mathbf{e}_i , \mathbf{s}_j and \mathbf{e}_j be start points and endpoints of line segments \mathbf{l}_i and \mathbf{l}_j , respectively. m and n represent the number of line segments in I_1 and I_2 . We define $\mathbb{C}_1 = \{\mathbf{s}_i, \mathbf{e}_i | i \in 1, 2, \dots, m\}$ and $\mathbb{C}_2 = \{\mathbf{s}_j, \mathbf{e}_j | j \in 1, 2, \dots, n\}$ as sets of line segments extracted from I_1 and I_2 , respectively. For a line segment \mathbf{l}_k , the direction vector of \mathbf{l}_k is denoted as:

$$\mathbf{l}_k = \frac{\mathbf{s}_k - \mathbf{e}_k}{\|\mathbf{s}_k - \mathbf{e}_k\|_2}. \quad (4)$$

As shown in Fig. 3, for a line segment $\mathbf{l}_i \in \mathbb{C}_1$, the predicted line in I_2 is \mathbf{l}'_i , and the correspondence is \mathbf{l}_j . The angle θ_{ij} between \mathbf{l}'_i and $\mathbf{l}_j \in \mathbb{C}_2$ is denoted as:

$$\theta_{ij} = \arctan \left(\frac{\|\mathbf{l}'_i \times \mathbf{l}_j\|_2}{\mathbf{l}'_i \cdot \mathbf{l}_j} \right). \quad (5)$$

For \mathbf{l}'_i , we sample h points on it. For a sampled point \mathbf{p}_w , the point-to-line distance between \mathbf{p}_w and \mathbf{l}_j is $\mathbf{p}_w^T \mathbf{l}_j$ with homogeneous coordinates. Based on the point-to-line distance, we denote the distance between \mathbf{l}'_i and \mathbf{l}_j as:

$$d_{ij} = \frac{1}{h} \sum_{x=0}^{h-1} \mathbf{p}_w^T \mathbf{l}_j. \quad (6)$$

Additionally, we take the midpoint distance between \mathbf{l}'_i and \mathbf{l}_j as:

$$m_{ij} = \left\| \mathbf{m}'_i - \mathbf{m}_j \right\|_2, \quad (7)$$

where $\mathbf{m}'_i = (\mathbf{s}'_i + \mathbf{e}'_i)/2$ and $\mathbf{m}_j = (\mathbf{s}_j + \mathbf{e}_j)/2$ are midpoints of \mathbf{l}'_i and \mathbf{l}_j . Then, we define $[\theta_{ij} \ d_{ij} \ m_{ij}]^T$ as geometric constraints between \mathbf{l}_i and \mathbf{l}_j .

For its candidates in I_2 , we calculate the geometric distance between \mathbf{l}_i and them. Then we construct the error matrix \mathbf{A}_i of \mathbf{l}_i . We denote \mathbf{A}_i as:

$$\mathbf{A}_i = \begin{bmatrix} \theta_{i0} & \theta_{i1} & \dots & \theta_{ij} & \dots & \theta_{i(n-1)} \\ d_{i0} & d_{i1} & \dots & d_{ij} & \dots & d_{i(n-1)} \\ m_{i0} & m_{i1} & \dots & m_{ij} & \dots & m_{i(n-1)} \end{bmatrix}_{3 \times n}. \quad (8)$$

Let x_{ij} denote a match flag between \mathbf{l}_i and \mathbf{l}_j . If $x_{ij} = 1$, \mathbf{l}_i and \mathbf{l}_j are a pair of corresponding line segments. If $x_{ij} = 0$, \mathbf{l}_i and \mathbf{l}_j are not correct correspondence. For $\mathbf{l}_i \in \mathbb{C}_1$, we define a n -dimensional vector $\mathbf{x}_i = [x_{i1} \ \dots \ x_{ij} \ \dots \ x_{in}]^T$ as the matching vector. Ideally, the matching vector should be equal to $\mathbf{x}_i = [0 \ \dots \ 1 \ \dots \ 0]^T$, where the index of number 1 indicates the correct correspondence of \mathbf{l}_i . Theoretically, the error matrix \mathbf{A}_i and the matching vector \mathbf{x}_i should satisfy the linear constraint

$$\mathbf{A}_i \mathbf{x}_i = \mathbf{b}, \quad (9)$$

where $\mathbf{b} = [0 \ 0 \ 0]^T$ represents the matching error, which should be equal to zero.

Generally, (9) can be solved easily by using the minimum ℓ^2 -norm solution. However, the solution of ℓ^2 -norm is generally dense [11]. In this paper, we want the matching vector \mathbf{x}_i be sufficiently sparse. Therefore, we adopt the ℓ^1 -norm minimization to ensure the sparsity of \mathbf{x}_i .

$$\min_{\mathbf{x}_i} \|\mathbf{x}_i\|_1 \quad s.t. \quad \|\mathbf{A}_i \mathbf{x}_i - \mathbf{b}\|_2^2 \leq \varepsilon, \quad (10)$$

where the constraint of (10) represents the geometrical constraints and ε denotes the maximum error that can be tolerated. Further, (10) can be converted into an unconstrained form

$$\min_{\mathbf{x}_i} \lambda \|\mathbf{x}_i\|_1 + \|\mathbf{A}_i \mathbf{x}_i - \mathbf{b}\|_2^2 / 2, \quad (11)$$

where λ is a weighting parameter.

Because ℓ^1 -norm minimization is convex, we can solve (11) based on [32]. Benefiting from the reliable predicted line segments by using the IMU-KLT algorithm, the proposed convex geometric distance minimization can converge accurately and efficiently. After obtaining \mathbf{x}_i , we normalize it and then take the position of its maximum element. If the position is denoted as j_0 , the j_0 -th line segment in the candidates is the correspondence of \mathbf{l}_i . For other line segments in I_1 , we repeat the above steps to obtain their correspondences in I_2 .

IV. POINT TRACKING METHOD

In this section, we go to obtain point correspondences. Remarkably, the proposed method is based on epipolar and image appearance constraints.

Let \mathbf{l}_i and \mathbf{l}_j be a pair of corresponding line segments in I_1 and I_2 , respectively. For a point \mathbf{p}_x sampled on \mathbf{l}_i , according to the epipolar constraint, we can get its epipolar line in I_2 easily. The intersection \mathbf{p}_y of the epipolar line and \mathbf{l}_j is denoted as:

$$\mathbf{p}_y = [\mathbf{F} \mathbf{p}_x] \times \mathbf{l}_j, \quad (12)$$

where \mathbf{F} is the fundamental matrix between I_1 and I_2 .

In practice, if the fundamental matrix \mathbf{F} is not accurate enough, then \mathbf{p}_y is not the correct correspondence of \mathbf{p}_x .

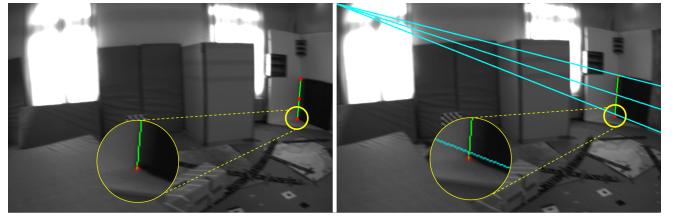


Fig. 4. Illustration of the proposed point tracking method. Green lines are a pair of corresponding line segments and the red points in left frame represent points that we want to track. The cyan lines in the right are epipolar lines of the red points. The yellow circles show that the fundamental matrix is inaccurate, which causes that the intersections are not correct correspondences, where the two bigger circle regions are the amplified parts of the two smaller ones. We use NCC score to search the correct positions of the red points along the corresponding line segment.

as depicted in Fig. 4. Therefore, it is necessary to search for a good correspondence of \mathbf{p}_x along \mathbf{l}_j based on image appearance constraint. We use 5×5 patches to describe \mathbf{p}_x and \mathbf{p}_y . We search for \mathbf{p}_y with the best normalized cross correlation (NCC) score. Empirically, the search window is set to ± 10 pixels. Besides, we reject false matches by a NCC score ($NCC < 0.90$). We search along \mathbf{l}_j rather than the epipolar line because this way is more reliable. The point tracking by the line tracking is more reliable. Then, some well-established point SLAM/VIO systems can be applied.

V. EXPERIMENTS

In this section, we evaluate the proposed line tracking algorithm on public datasets. We compare our line tracking method with the popular descriptor-based approach, LBD [13], with two different line segment detectors, LSD [27] and FLD [28]. We have given the reasons why we compare with LBD [13] in Section II. Besides, to further evaluate the proposed line tracking method, we integrate it into OpenVINS [1]. Then the system is tested on the EuRoC dataset [31] and a home-made dataset. All experiments have been run in an Intel Core i7-CPU, at 3.2GHz, with 32GB RAM. For all experimental results, we report the median of five times.

A. Line Segment Tracking Evaluation

We apply LSD [27] and FLD [28] to detect line segments and examine the tracking algorithms of LBD [13] and our proposed method. We select two public datasets, the TUM-RGBD [29] and the EuRoC MAV dataset [31]. Normally, the proposed line segment tracking method fuses IMU information to aid the KLT algorithm, where when the IMU information is not available, the rotation matrix in Section III-A is set to the identity matrix.

In order to evaluate the algorithms quantitatively, we use projection errors to classify line segments like [11] and [16]. Empirically, we classify each match as an inlier if the projection error is less than five pixels. In the experiments, we use the number of correct matches and inlier ratio to assess the algorithms. The number of correct matches means average correct line segment matches and the inlier ratio

TABLE I
TRACKING PERFORMANCE OF THE PROPOSED METHOD AND LBD-BASED LINE SEGMENT TRACKING METHOD
(NUMBER REPRESENTS NUMBER OF CORRECT MATCHES AND RATIO REFERS TO INLIER RATIO)

Dataset	LSD+LBD		LSD+Ours		FLD+LBD		FLD+Ours	
	number	ratio	number	ratio	number	ratio	number	ratio
tum-rgbd/fr1/360	61	61%	48	94%	35	39%	40	95%
tum-rgbd/fr1/desk	61	61%	56	94%	37	37%	51	95%
tum-rgbd/fr1/floor	63	63%	57	99%	42	43%	56	99%
tum-rgbd/fr1/room	66	66%	60	95%	42	42%	55	95%
tum-rgbd/fr2/360.hemisphere	71	71%	70	96%	44	46%	62	96%
tum-rgbd/fr2/360.kidnap	67	67%	66	95%	43	49%	61	95%
tum-rgbd/fr2/large.with.loop	75	75%	75	96%	48	49%	69	96%
EuRoC/Machine Hall 01	78	78%	74	95%	58	58%	71	95%
EuRoC/Machine Hall 02	77	77%	73	95%	58	58%	70	95%
EuRoC/Vicon Room1 01	80	80%	74	97%	59	59%	72	97%
EuRoC/Vicon Room2 01	83	83%	76	97%	61	62%	72	97%

refers to the ratio of the number of correct matches to the total number of matches in a sequence.

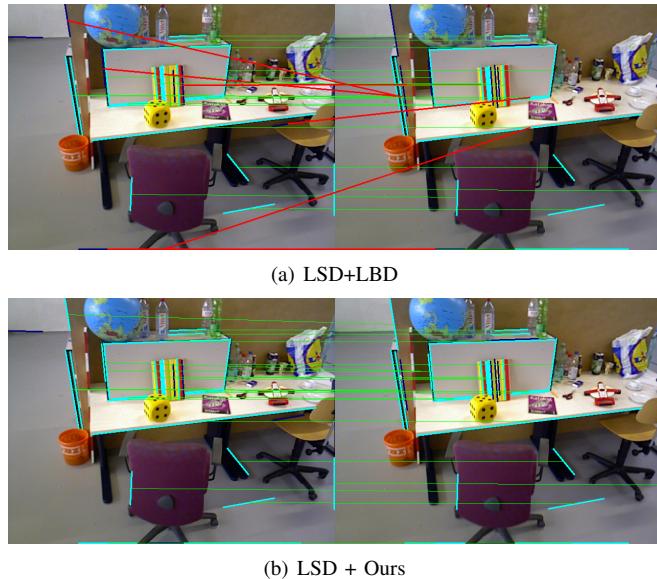


Fig. 5. Line tracking results of ours and LBD [13] using LSD [27] line detector. The image pairs are from the TUM-RGBD dataset [29]. Red lines represent false matches by LBD-based method.

In Table I, we report the average number of correct matches and the inlier ratio of sequences from the TUM-RGBD dataset [29] and the EuRoC dataset [31]. First of all, in terms of inlier ratio, when using LSD [27] to detect line segments, the proposed method achieves 95.71% in average, and LBD [13] is 71.1%. When using FLD [28], the proposed method achieves 95.89% in average, while LBD [13] is only 49.21%. The main reason is that the proposed method benefits from the strict geometric constraints. Second, in terms of the number of correct matches, when using FLD [28] to detect line segments, the proposed method has a better performance than LBD [13]. The main reason accounts for the fact that FLD [28] is based on image edges and the proposed method also depends on image edges. When using LSD [27] to detect line segments, the proposed method has a slightly inferior performance compared with LBD [13]. This

TABLE II
COMPARISON OF THE COMPUTATIONAL COST FOR DIFFERENT ALGORITHMS ON THE TUM-RGBD DATASET [29]

Method	LSD+LBD	LSD+Ours	FLD+LBD	FLD+Ours
Extraction	15.8 ms	15.8 ms	3.5 ms	3.5 ms
Tracking	17.9 ms	3.8 ms	17.9 ms	3.7 ms
Total	33.7 ms	19.5 ms	21.4 ms	7.2 ms

is because the proposed method uses the strict geometric constraints to restrict line segment tracking. In summary, for the proposed method, it is better to utilize FLD [28] to detect line segments.

In Table II, we compare the computational cost of the algorithms. Obviously, the proposed algorithm is faster than others. We see that the proposed algorithm with FLD [28] only takes 7.2 ms to process all tasks, while the LBD [13] with FLD [28] takes 21.4 ms to process a frame. For line extraction, FLD [28] is much efficient than LSD [27]. For line tracking, the proposed method is much faster than the LBD-based method.

Intuitively, Fig. 1 and Fig. 5 show line segment tracking results of the LBD-based [13] method and ours. In Fig. 1, we use FLD [28] to detect line segments. Differently, in Fig. 5, we use LSD [27]. As shown in Fig. 1 and Fig. 5, our method achieves better matching performances than the LBD-based method.

In summary, the proposed line segment tracking with FLD achieves much higher accuracy and simultaneously much higher speed. The above experiments show that not only the number of correct matches increases but also the inlier ratios are increased by at least 35.1% and up to 57.3% from state-of-the-art level. Moreover, the proposed line segment tracking is three times faster.

B. Evaluation on VIO

In this part, to further evaluate the proposed line segment tracking method, we adapt it into a point-based VIO system, OpenVINS [1], and we use FLD [28] to detect line segments. We evaluate the system with the proposed line segment tracking on the EuRoC dataset [31] and a home-made dataset.

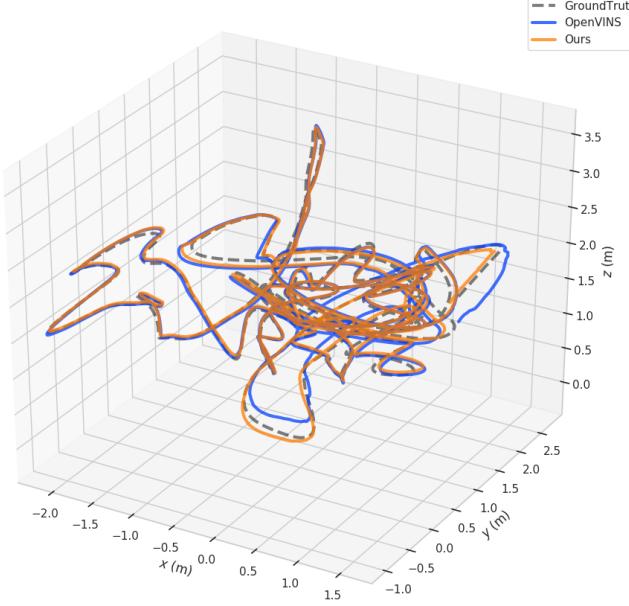


Fig. 6. Comparision of the trajectories by using the OpenVINS [1] and the adapted OpenVINS in V1_03 of the EuRoC dataset [31]. The gray dashed line shows the ground truth, the blue line is the trajectory obtained by OpenVINS, and the orange line is the trajectory obtained with the adapted OpenVINS (Ours).

TABLE III
RELATIVE RMSE ERRORS ON THE EUROC MAV DATASET [31]
(IN METERS)

Sequence	OpenVINS	Adapted OpenVINS
Machine Hall 01	0.184	0.172
Machine Hall 02	0.160	0.156
Machine Hall 03	0.127	0.130
Machine Hall 04	0.172	0.147
Machine Hall 05	0.343	0.332
Vicon Room 1 01	0.080	0.067
Vicon Room 1 02	0.118	0.073
Vicon Room 1 03	0.207	0.111
Vicon Room 2 01	0.102	0.100
Vicon Room 2 02	0.059	0.061
Vicon Room 2 03	0.178	0.176

EuRoC Dataset. Table III shows the relative RMSE errors of the trajectories obtained by OpenVINS [1] and the adapted OpenVINS. We can observe that the adapted OpenVINS achieves better precision in most of the sequences, which is benefited from integrating the proposed line segment tracking method. The reason is that there are more effective observations by the adapted OpenVINS than by the original one. In Fig. 6, the trajectories are aligned with the groundtruth. We see that accuracy is improved by the adapted OpenVINS.

Home-made Dataset. Besides, we examine the adapted OpenVINS in a home-made dataset. We compare the adapted OpenVINS with a state-of-the-art method, ORB-SLAM3 [3] in monocular-inertial mode. Experimental results show that the adapted system performs better than ORB-SLAM3. Fig. 7 (a) shows an indoor scene, where the shape of stairs obtained

by the adapted OpenVINS is clear, while ORB-SLAM3 fails due to lack of reliable feature points. In a rapid rotation situation like Fig. 7 (b), ORB-SLAM3 fails due to a lot of incorrect 3D-2D correspondences by using ORB descriptors for point matching. We recommend watching the attached video for more details and more comparisons.

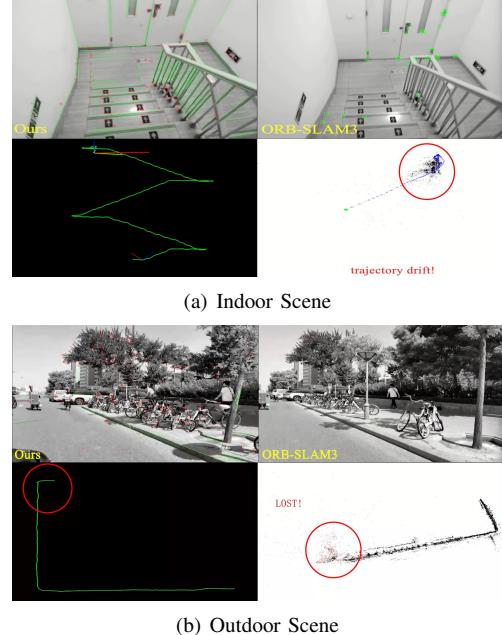


Fig. 7. Results of ORB-SLAM3 [3] and the adapted OpenVINS.

VI. CONCLUSIONS

In this paper, we propose an accurate and fast line segment tracking method. The method utilizes an IMU-aided KLT algorithm to predict line segments, which can offer reliable initial values for the line segment tracking. Furthermore, a novel geometric distance between two line segments is proposed to measure the predicted line segments and extracted line segments. We model the line segment tracking as a ℓ^1 -norm convex minimization. To observe the behaviors of the line segment tracking in a VIO system, we provide a reliable point tracking method based on epipolar and appearance constraints. We perform experiments on public datasets and a dataset captured by ourselves. On the TUM-RGBD dataset [29], the proposed line segment tracking only takes 7.2 ms to process a frame, which is 3 times faster than state-of-the-art level. More importantly, it achieves inlier ratios greater than 94%, which outperforms state-of-the-art level by at least 35.1% and up to 57.3%. We integrate the proposed line segment tracking algorithm into OpenVINS [1]. The adapted OpenVINS is improved in terms of accuracy.

In the future, we will employ more geometric features like edges and planes in SLAM systems to enhance accuracy and robustness in more challenging environments.

ACKNOWLEDGMENT

This work is supported by the Open Projects Program of National Laboratory of Pattern Recognition 202100040.

REFERENCES

- [1] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, “Openvins: A research platform for visual-inertial estimation,” in *IROS 2019 Workshop on Visual-Inertial Navigation: Challenges and Applications*, 2019.
- [2] J. Engel, V. Koltun, and D. Cremers, “Direct sparse odometry,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.
- [3] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, “Orb-slam3: An accurate open-source library for visual, visual-inertial and multi-map slam,” *arXiv preprint arXiv:2007.11898*, 2020.
- [4] G. Zhang, J. H. Lee, J. Lim, and I. H. Suh, “Building a 3-d line-based map using stereo slam,” *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1364–1377, 2015.
- [5] R. Gomez-Ojeda, F.-A. Moreno, D. Zuñiga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez, “Pl-slam: A stereo slam system through the combination of points and line segments,” *IEEE Transactions on Robotics*, vol. 35, no. 3, pp. 734–746, 2019.
- [6] S.-J. Li, B. Ren, Y. Liu, M.-M. Cheng, D. Frost, and V. A. Prisacariu, “Direct line guidance odometry,” in *2018 IEEE international conference on Robotics and automation (ICRA)*. IEEE, 2018, pp. 1–7.
- [7] S.-S. Huang, Z.-Y. Ma, T.-J. Mu, H. Fu, and S.-M. Hu, “Lidar-monocular visual odometry using point and line features,” in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 1091–1097.
- [8] Q. Wang, Z. Yan, J. Wang, F. Xue, W. Ma, and H. Zha, “Line flow based slam,” *arXiv preprint arXiv:2009.09972*, 2020.
- [9] B. Fan, F. Wu, and Z. Hu, “Robust line matching through line-point invariants,” *Pattern Recognition*, vol. 45, no. 2, pp. 794–805, 2012.
- [10] J.-Y. Bouguet *et al.*, “Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm,” *Intel corporation*, vol. 5, no. 1–10, p. 4, 2001.
- [11] R. Gomez-Ojeda and J. Gonzalez-Jimenez, “Geometric-based line segment tracking for hdr stereo sequences,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 69–74.
- [12] Z. Wang, F. Wu, and Z. Hu, “Msld: A robust descriptor for line matching,” *Pattern Recognition*, vol. 42, no. 5, pp. 941–953, 2009.
- [13] L. Zhang and R. Koch, “An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency,” *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 794–805, 2013.
- [14] M. Lange, F. Schweinfurth, and A. Schilling, “Dld: A deep learning based line descriptor for line feature matching,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5910–5915.
- [15] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [16] Q. Ma, G. Jiang, and D. Lai, “Robust line segments matching via graph convolution networks,” *arXiv preprint arXiv:2004.04993*, 2020.
- [17] C. Forster, M. Pizzoli, and D. Scaramuzza, “Svo: Fast semi-direct monocular visual odometry,” in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 15–22.
- [18] T. Qin, P. Li, and S. Shen, “Vins-mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [19] F. Schenk and F. Fraundorfer, “Reslam: A real-time robust edge-based slam system,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 154–160.
- [20] S. J. Lee and S. S. Hwang, “Elaborate monocular point and line slam with robust initialization,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1121–1129.
- [21] S. Yang and S. Scherer, “Monocular object and plane slam in structured environments,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3145–3152, 2019.
- [22] X. Zuo, X. Xie, Y. Liu, and G. Huang, “Robust visual slam with point and line features,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1775–1782.
- [23] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, “Pl-slam: Real-time monocular visual slam with points and lines,” in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 4503–4508.
- [24] S. Yang and S. Scherer, “Direct monocular odometry using points and lines,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3871–3877.
- [25] H. Zhou, D. Zou, L. Pei, R. Ying, P. Liu, and W. Yu, “Structslam: Visual slam with building structure lines,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1364–1375, 2015.
- [26] D. Zou, Y. Wu, L. Pei, H. Ling, and W. Yu, “Structvio: visual-inertial odometry with structural regularity of man-made environments,” *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 999–1013, 2019.
- [27] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, “Lsd: A fast line segment detector with a false detection control,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 4, pp. 722–732, 2008.
- [28] J. H. Lee, S. Lee, G. Zhang, J. Lim, W. K. Chung, and I. H. Suh, “Outdoor place recognition in urban environments using straight lines,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 5550–5557.
- [29] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of rgbd slam systems,” in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.
- [30] M. Chandraker, J. Lim, and D. Kriegman, “Moving in stereo: Efficient structure and motion using lines,” in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 1741–1748.
- [31] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, “The euroc micro aerial vehicle datasets,” *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.
- [32] D. L. Donoho and Y. Tsaig, “Fast solution of ℓ_1 -norm minimization problems when the solution may be sparse,” *IEEE Transactions on Information Theory*, vol. 54, no. 11, pp. 4789–4812, Nov 2008.