STRUCTURAL LINE FEATURE SELECTION FOR IMPROVING INDOOR VISUAL SLAM

Rui Xia, Ke Jiang, Xin Wang*, Zongqian Zhan

School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China
2017301610289@whu.edu.cn, jk0304@qq.com, (xwang, zqzhan)@sgg.whu.edu.cn

Commission IV, WG IV/5

KEY WORDS: Visual SLAM, Vanishing Points, Manhattan World Assumption, Structural Line Feature Selection

ABSTRACT:

Nowadays, Visual SLAM has gained ample successes in various scenarios. For feature-based system, it is still limited when running in an indoor room, as the indoor scene is often with few and simple texture which result in less and unevenly distributed point features. To solve this limitation, line features which are quite rich in an indoor scene are extracted and used. However, not all features can geometrically contribute to pose estimation, specifically, line features that are consistent to the motion direction provide only weak geometric constraint for solving pose parameters. Therefore, this paper proposes a selection method for reasonable line features, in particular, based on the Manhattan World Assumption (MWA), structural line features are firstly extracted instead of normal line features. Then, the structural line features are selected according to the direction information of vanishing points and selected for a stronger geometric constraint on pose estimation. In general, the selected structural lines require that the intersection angle between the corresponding principal direction and the camera motion direction is higher than a threshold, which is extensively investigated in the experiments. The experimental results show that, compared to the original ORB-SLAM2, the localization accuracy after using the proposed method can be improved by around 15%-40% on various public datasets, and the real-time performance can be basically guaranteed even including the extra time spent on the selection procedure.

1. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) plays an important role in many fields, such as autonomous driving, robot navigation, augmented reality and etc. On the other hand, as the technique of sensors develops, it becomes more and more convenient and practical to acquire data, such as images, videos, conventionally, it is called Visual SLAM if the input are frames. Typically, a Visual SLAM framework often contains two procedures: front-end and back-end, in which the front-end is also known as visual odometry for continuously tracking the sensor and estimating the corresponding pose and the back-end aims to optimize the both the localization and mapping information including loop closure refinement.

In the front-end part, one of the most popular methods is based on extracting local features, which is commonly called feature-based method (accordingly, another one is called direct method (Forster et al., 2014), which is not relevant to this paper and not introduced here). The feature-based method extracts local point features (ORB (Rublee et al., 2011), Harris (2014)) and corresponding descriptor for feature tracking and matching, and estimate camera pose. Thanks to the characteristics of various point features, the feature-based method is the mainstream method in Visual SLAM as it is relatively more stable and insensitive to illumination and dynamic objects. However, in some real applications, the feature-based methods have limitations in some specific cases, for example, in indoor scenario where the main environment is planar and linear, and the number of point features is relatively sparse and unevenly distributed when the texture is not very rich, thus, it is difficult to have high robustness and accuracy by just using point features. To cope with this limitation, researchers have tried to introduce additional features as observations in the front-end, in which the line feature is one of the most common solutions. In 2016, PL-SVO (Gomez-Ojeda et al., 2016) was proposed to integrate line features into a lightweight semi-direct visual odometry, the approach only works on the problem of visual odometry as loop closure is not considered. PL-SLAM (Gomez-Ojeda et al., 2019) is a stereo Visual SLAM that uses the combination of point and line features in the procedure of BA (*Bundle Adjustment*, Triggs et al., 1999) optimization and loop closure detection, in this work all the extracted line features are used. The StructSLAM (Zhou et al., 2015) employed the structural line features instead of all line features and EKF (Extended Kalman Filter) for back-end optimization, it was shown with good performance regarding the location precision, but, due to the inherent limitations of EKF, it is difficult to deal with long-term frames. The StructSLAM also demonstrated that the structural line features have stronger geometric constraint when estimating the pose of sensors than applying all line features does, the BA-based optimization can handle larger scale data than the EKF-based optimization does. In general, the advantage of using line features for SLAM is that lines contain more environmental information than points, and in scenes such as indoor scenes that lack point features, line features can be introduced.

Generally, in a real indoor environment, line features have properties of multi-directions and structure. Unlike point feature with a specific localization, line features are always with extra direction information, and different line features are in different directions according to the observed scenes. Line features with different directions are in effect not equally important for solving camera poses, e.g., the line features which are just parallel to the direction that camera rotates around or parallel to the camera moving direction, these line features provide relatively weak geometric constraint for pose estimation. On the other hand, the structure of an indoor scenario is often regular and can be seen as a superposition of rectangular blocks with three principal directions, in line with the Manhattan World Assumption (Coughlan et al., 1999). These lines in the principal direction are called structural lines (as Fig. 1 shows), the difference between structural lines and normal lines is that structural lines contain structural information of the observed scene, which can improve the estimation of camera poses (Han et al., 2021). In addition, structural lines can outline the indoor environment in a more representative way. Motivated by the

* Corresponding author.

introduced two properties, to improve the performance of Visual SLAM in an indoor environment, this paper investigates the influence of various line features' direction information on pose estimation and present a selection method for robust structural line features, more specifically, the structural line features are considered instead of using all normal line features to guarantee the real-time performance, and structural line features that are more geometrically reliable are selected for subsequent pose estimation.

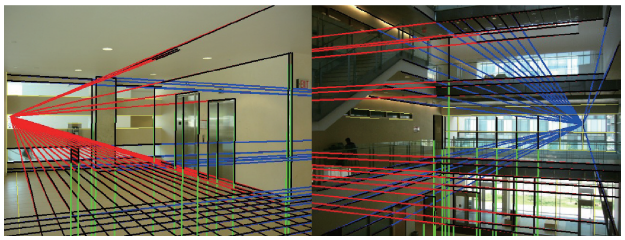

Figure 1. Structural lines in an indoor environment. The structural lines often have strong structural regularity, which are consistent with the Cartesian coordinate system, and can be extended to three principal directions (Han et al., 2021).

Our main contributions are *threefold*: First, the influence of directions of various line features are analyzed; Second, based on the camera moving direction, we proposed a structural line feature selection method for improving the indoor Visual SLAM; Third, the most reasonable free parameter is advocated by extensive experiments.

The rest of this paper is organized as follows. Related works are reviewed in Section 2. The details of our method are explained in Section 3. The performance of our works on different datasets and the experimental results are reported in Section 4. Finally, conclusions and an outlook are drawn in Section 5.

2. RELATED WORK

There have been many research works to improve indoor Visual SLAM by introducing line features. In this section, we briefly review the studies of Visual SLAM methods. As there are a great many SLAM research works in fields of photogrammetry, computer vision, robotics etc., in this section, among which, two categories of methods that are relevant to our work are discussed: point-line fusion and structural line features.

2.1 Visual SLAM with point-line fusion

Most Visual SLAM methods use only point features as input to estimate the camera pose and build the map of the surrounding environment, such as PTAM (Klein et al., 2007) with BA (bundle adjustment) method on the back-end and MonoSLAM (Davison et al., 2007) with EKF filter as the back-end. The advantage of these point feature-based SLAM methods is that point features are easy to extract and track, while one disadvantage is that for some man-made environments such as corridors of walls, the accuracy of SLAM is often negatively affected by sparse feature points, such as Fig. 2 illustrates. In addition, there are also many works to implement SLAM using line segments, and most of them adopt different parameterization methods for line features, in fact they still represent line feature by two endpoints of a line segment and track this line feature by tracking the corresponding two endpoints, which is similar to the previous point-feature-based SLAM. Sola et al. (2009) provided a comprehensive investigation of different parameterization methods of point features and line features and extensive experiments were reported by comparing these point and line features. There have also been many attempts to use vanishing points to improve accuracy, Ma et al. (2019) designed a cost function to minimize both of the reprojection error of line segments and alignment error of the vanishing points in the back-end module, Lim et al. (2022) proved that vanishing point measurements guarantee a unique mapping solution through Fisher information matrix rank analysis. Chandraker et al. (2009) use stereo cameras to generate line features, they used straight lines instead of line segments to design an efficient system that performs robustly in complex indoor environments. PL-SVO (Gomez-Ojeda R. et al., 2016) is an earlier method that adds line features to a lightweight semi-direct visual odometry, this method is not a complete Visual SLAM system due to the lack of a loop closure detection module. One year later, a more complete Monocular PL-SLAM (Pumarola et al., 2017) and Stereo PL-SLAM (Gomez-Ojeda et al., 2017) methods were proposed, the former is built upon ORB-SLAM which can work even if most of the features points are vanished out from the input images; the latter contributed a novel bag-of-words approach that exploited the combined the two kinds of point and line features in loop-closure procedure, and the resulting map is denser and more diverse in three-dimensional elements.

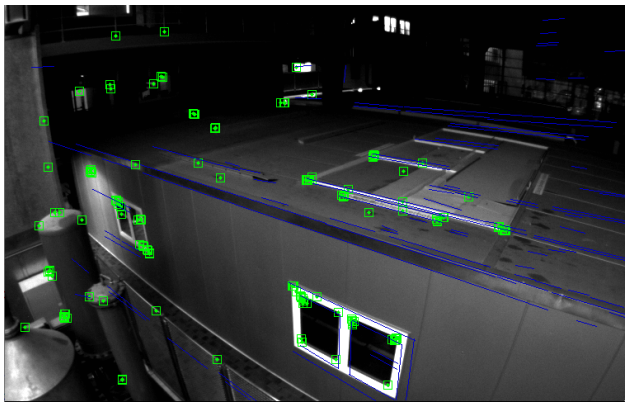

Figure 2. Point-line fusion tracking. When the indoor scene has fewer feature points (green markers in the figure) extracted due to factors such as lighting and texture, line features (blue line markers in the figure) are introduced as additional observations.

Line features, as a higher-level structural representation, have been widely applied by many researchers to improve the performance of Visual SLAM thanks to the fact that more information about the environment is included. However, in real cases, line features are very complex and not stable to extract, therefore, a simple line feature that takes the regularity of environment into consideration is studied, i.e., structural line features.

2.2 Structural line features

The concept of structural lines was introduced by Coughlan (1999), according to the Manhattan World Assumption, most man-made buildings have three main planes perpendicular to each other, and the straight lines parallel to the main plane directions can outline the general structure of the building, which are called the structural lines. Unlike the normal line feature, the structural line feature typically contains the whole structural information of the scene, which can contribute an overall control during each step of pose estimation and reduce the cumulative error, thus also improve the accuracy of pose estimation. As shown in Fig. 3, the indoor environment has three principal directions, one vertical direction and two horizontal directions that are perpendicular to each other. There are three groups of structural lines based on three principal directions, which are marked by different colors. Each group of structural line has a corresponding principal direction, which can be represented by a corresponding vanishing point. Since the structural lines contain the orientation information of the overall scene structure, they have a strong geometric control and can make more environment scene

information be involved into the pose estimation of the SLAM.

Based on this structural line feature, Zhou et al. (2015) proposed the StructSLAM and used EKF (Extended Kalman Filter) for back-end optimization, and their experiments demonstrated that applying structural line feature could have stronger geometric constraints than applying all line feature when estimating the camera pose.

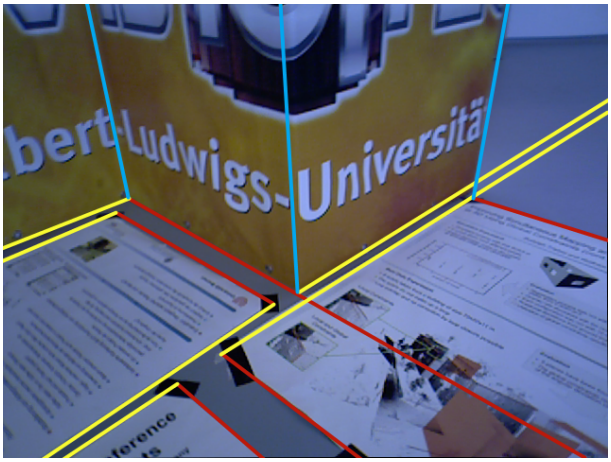

Figure 3. Structural lines of an indoor scene. The indoor scene generally contains three principal directions, and the directions of the structural lines (indicated in yellow, blue and red in the figure) are parallel to these three principal directions.

The above reviewed works that utilize line features consider all extracted line features with equal influence on determining camera pose. However, due to the multi-direction property of line features, different directional line features have different influence when using the corresponding geometric constraint.

Our method selects the line features by only considering structural line features based on the direction of camera moving and structural line features. From a geometric perspective, we explore the influence of structural line features with various direction on estimating camera pose, and these structural line features that are more helpful to enhance the geometric solution are selected.

3. METHODOLOGY

3.1 Overview of our developed method

The implemented framework is based on the open package ORB-SLAM2, the related working pipeline integrating with our method is illustrated in Fig. 4. In general, our whole working pipeline is consistent with the original package, which contains multi-threads for dealing with tracking, mapping and optimization. The relevant upgrade by our method is concentrated on the tracking part as the red dashed box shows, basically, based on the MWA, structural line features are firstly generated and used instead of all extracted normal line features, the principal direction of each structural line features are then determined after estimating vanishing point, camera motion direction is also computed given the camera motion model, finally, these direction information are used to select robust structural line features for pose estimation. More methodological details are introduced in the next subsections, basics of line features are briefly stated in section 3.2, the influence of line feature direction on pose estimation is explained in section 3.3, details of structural line feature selection are shown in section 3.4.

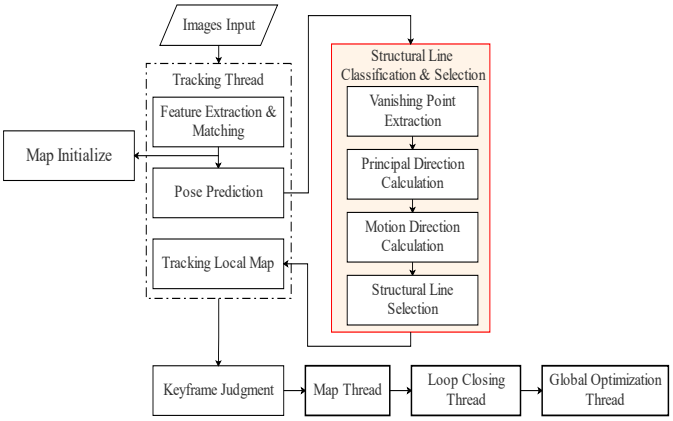

Figure 4. Flowchart of the method in this paper.

3.2 The extraction, description and parameterization of line features

In this paper, line features are extracted using the LSD (*Line Segment Detector*, Grompone von Gioi et al., 2010) algorithm whose main idea is based on pixel clustering and has a good performance with high accuracy and real-time capability. In addition, the LBD (*Line Band Detector*, Zhang et al., 2013) descriptor is used to describe the line features which employs the pixel information of the line feature neighborhood to ensure a robust matching result. The LBD explore scale space by inspecting various pyramid images and the generated descriptor is robust to scale changes, one selling point is that the LBD descriptor is very fast to calculate.

For our work, the Prücker Coordinate (Bartoli et al., 2003) and the Orthogonal (Bartoli et al., 2005) are introduced to parameterize the line features. The Prücker Coordinate is a six-parameter representation which uses two vectors to represent spatial straight lines and can both easily and intuitively perform spatial rigid body transformations in the following form:

$$L_{Prücker} = (n, d) \tag{1}$$

where $L_{Prücker}$ represents the Plückian Coordinates, n represents the normal vector of the plane formed by the line and the origin, and d represents the direction vector of the line.

The Orthogonal, which can be derived from the Prücker Coordinate, is a four-parameter representation and consistent with the degrees of freedom of a spatial straight line and can therefore be used for back-end BA optimization in the following form:

$$L_{Orthogonal} = (U, W)_{U \subseteq SO3, W \subseteq SO2} \tag{2}$$

where

$$U = [\Psi_1 \quad \Psi_2 \quad \Psi_3] = \left[\frac{n}{\|n\|} \quad \frac{d}{\|d\|} \quad \frac{n \times d}{\|n \times d\|}\right] \tag{3}$$

$$W = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \frac{1}{\sqrt{(\|n\|^2 + \|d\|^2)}} \begin{bmatrix} \|n\| & -\|d\| \\ \|d\| & \|n\| \end{bmatrix} \tag{4}$$

The relationship between the Prücker Coordinate and the Orthogonal is as follows:

$$L_{Orthogonal} = \frac{1}{\sqrt{(\|n\|^2 + \|d\|^2)}} L_{Prücker} \tag{5}$$

where $L_{Orthogonal}$ is the line in the Orthogonal, $L_{Prücker}$ is the line in the Prücker Coordinate.

3.3 Influence of line feature direction on pose estimation

According to formula (1), the Prücker Coordinate considers a line feature in the image plane as an infinite straight line in the corresponding space, which can leads to that when the camera rotates around the axis orthogonal to the line feature direction or translates along the line feature direction, it does not change the position projected on the image, and the reprojection error of this straight line in object space cannot reflect a valid geometric constraint on the camera's pose estimation (Pumarola et al., 2017).

A qualitative example is given by in Fig. 5, L is a spatial straight line, O is the projection center of the camera, and $A1$, $A2$, $B1$, $B2$ are the points on the spatial straight line. The camera observes $A1$ and $B1$ at a certain position, and the re-projections on the image are $a1$, $b1$. Assume that the camera rotates a bit around the axis that is orthogonal to the direction of the line L, it observes $A2$ and $B2$, and the re-projections on the image are $a2$ and $b2$, it can be easily figured out that $a1b1$ and $a2b2$ are the same line l, and they contribute to the same constraint when the reprojection error calculation is performed. Similarly, a similar situation occurs when the camera is translated along the straight line L. To cope with this degeneration, after extracting the structural lines, it is necessary to further select the extracted structural lines according to the motion trend of the camera.

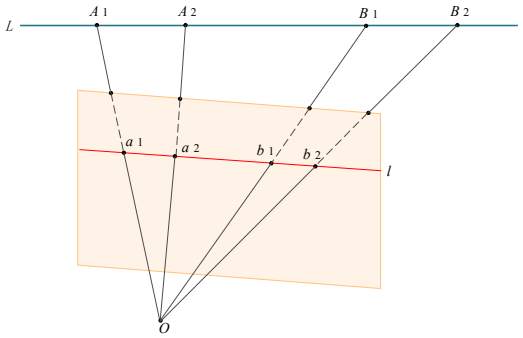

Figure 5. Spatial line reprojection model.

3.4 Selection of structural line features

According to the Manhattan World Assumption, the directions of the structural lines should coincide with the three principal directions of the observed scene and the direction of the vanishing point in the principal directions as well. Therefore, the structural line features can be selected according to the direction of the vanishing point.

To generate the structural line features, we employ the method of Lu et al. (2017) according to the vanishing point direction. As Fig. 6 shows, it mainly has four steps: building a polar grid, generating the vanishing point hypothesis, verifying the vanishing point hypothesis, and extracting structural line features based the verified vanishing points.

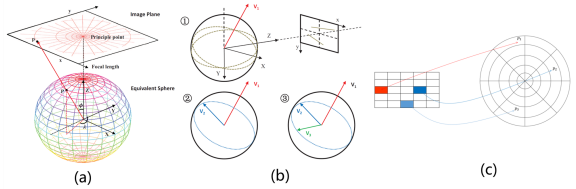

Figure 6. Schematic diagram of the process of generating structural line features. (a) polar grid building; (b) vanishing point hypothesis generating; (c) vanishing point hypothesis verifying. (Lu et al., 2017)

After extracting the structural lines, from the previous discussion, the structural line features whose direction vectors are orthogonal to the current motion direction vector should with high priority to be selected. Similarly, according to the Manhattan World Assumption, we use the directions of the vanishing points to represent the directions of the structural line features in the corresponding principal directions. In this paper, the structural line features are selected by the following three steps:

1. Calculating principal directions. The vanishing points are projected onto the 3D space according to the initial pose estimation, then the principal directions in which the vanishing points lie are calculated.
2. Estimating camera motion direction. Obtain the current relative camera motion direction according to the assumed motion model with consistent velocity.
3. Selecting structural line features. The angular difference between the structural line feature direction and the motion direction vector is first calculated, and then, the structural line is supposed to be the selected candidate if the corresponding angular difference is greater than a threshold α.

4. EXPERIMENTS

To demonstrate the efficacy of our proposed method, we conducted two experiments on two public indoor datasets, EuRoC[1] and TUM[2]. One is to comprehensively explore the influence of different angle thresholds α on the performance of the proposed method. The second experiment is an ablation study by comparing different variants among the methods in this paper. Both experiments mainly evaluate two metrics of time consuming and accuracy on different datasets. All experimental results were generated by an operating environment with six 3.20GHz Intel Core i7-8700 processors and 12 threads.

In addition to the proposed method (the results of our method are indicated as *our*), the ablation studies in this part are:

1. The original ORB-SLAM2 method. The results are indicated as *ORB-SLAM2*.
2. The original monocular PL-SLAM method (using traditional line features). The results are indicated as *NomalLine*.
3. The relevant method using structural line features which are parallel to camera motion direction. The results are indicated as *ParallelLine*.

In the following subsections, we first describe the experimental datasets, and introduce the metrics for evaluation. Then, the corresponding two experiments are shown with more details.

4.1 Experimental Datasets

EuRoC Dataset: We selected three sequences, *MH01*, *MH05* and *V203*, from the EuRoC dataset. As Fig. 7 and Fig. 8 show, the observed scenes of these three sequences are all from indoor environment and basically is identical with the Manhattan World Assumption, and in terms of tracking difficulty, there are both easy sequence (*MH01*) and difficult sequences (*MH05* and *V203*), so that our method can be comprehensively evaluated. The EuRoC dataset contains two scenes, one is a machine hall at the Zurich ETH, and the other one is a common room. The camera vehicle is an Asctec Firefly hex-rotor helicopter with a stereo VIO camera. The dataset provides ground-truth of the flight trajectories that have been

[1] More details related to EuRoC can be found at https://projects.asl.ethz.ch/datasets/doku.php?id=kmavvisualinertialdatasets.
[2] More details related to TUM can be found at https://vision.in.tum.de.

spatial-temporally aligned.

Figure 7. Scene (machine hall at the Zurich ETH) from the *MH01* sequence (easy) and *MH05* sequence (difficult).

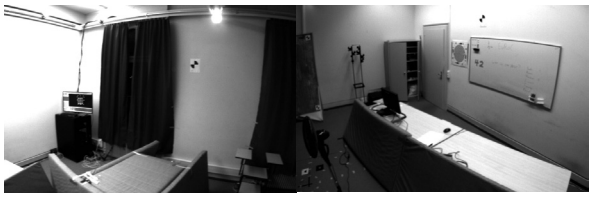

Figure 8. Scene (common room) from the *V203* sequence (difficult).

TUM Dataset: TUM is a series of datasets for computer vision, image processing and pattern recognition tasks, among which we use the *fr1_desk* (Fig. 9) and *fr3_str_tex_near* (Fig. 10) sequences, both of these two sequences are also indoor scenes and basically within the Manhattan World Assumption. *fr1_desk* contains more objects and details, but the camera swings relatively large, which can be used to test the stability of the proposed method on the whole ORB-SLAM2; *fr3_str_tex_near* is relatively simple and focuses on the performance of the SLAM method in terms of environmental structure and texture. The dataset was collected by Kinect and constitute RGB images and depth data, and also uses a high-precision motion acquisition device to acquire the real-time pose of the camera as the ground-truth for quantitative evaluation.

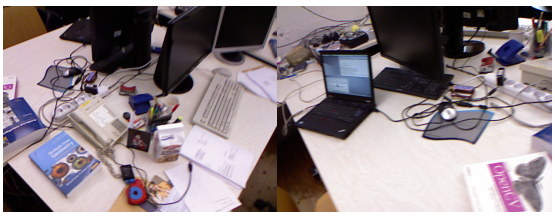

Figure 9. Sample images of *fr1_desk* sequence.

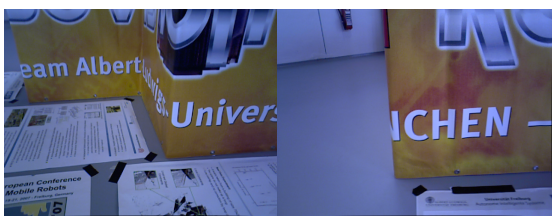

Figure 10. Sample images of *fr3_str_tex_near* sequence.

4.2 Evaluation metrics

We evaluate a SLAM method from two aspects: time-consuming and accuracy. Obviously, the metric of time-consuming reflects the computational complexity of the method. If the time-consuming is too high that the current frame has not been solved when the next frame arrives, the method is considered to be not able to meet the real-time requirements. In terms of accuracy, since only monocular Visual SLAM method is used in this paper, the scale of the output trajectory is uncertain, therefore, we use evo-tool [1] to align the generated trajectory with the coordinate system of ground truth according to the Least Squares, and then calculate the least squares trajectory error between the aligned trajectory and ground truth, the RMSE (*Root Mean Square Error*) value of the trajectory error is referred as the final accuracy, which is calculated as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_{res}^{i} - y_{GT}^{i}\right)^{2}} \tag{6}$$

where y_{res}^{i} represents the pose at moment i after trajectory alignment and y_{GT}^{i} represents the ground truth pose at moment i.

4.3 Experiment 1: Exploring the effect of angle threshold α

The value of the threshold α determines how many structural lines can be eliminated. Theoretically, the larger the threshold α is, the less likely it is that the angle between the line feature and the direction of camera motion is greater than the threshold, and also the more lines will be eliminated, this can result in fewer lines being kept in subsequent operations, and the time-consuming will be less. In terms of accuracy, if the threshold α was set too large, then the selected structural lines will be few or even equal to zero, and the trajectory error may gradually increase to approach the original ORB-SLAM2.

To demonstrate the described influence of the threshold α, based on the sequence of *V203*, we ran several experiments by testing eight different thresholds α with value of from 10 degrees to 80 degrees (interval of 10 degrees), and applied each of these eight different angle thresholds to the proposed method – *our*. Remind that the results of most SLAM systems appear to have some randomness, so we repeat the method for 5 times, and calculate the corresponding average time-consuming and average accuracy as the final evaluation metrics, as shown in Tab. 1.

It is clear from the table that if the threshold α is increased, then the time-consuming becomes less and the accuracy increases and then decreases, which is in line with our theoretical analysis. Moreover, as can be seen in Fig. 11, the number of selected structural lines does decrease as the angle threshold α increases. In conclusion, the accuracy is highest when the angle threshold α is about 50 degrees.

Table 1. The results of different thresholds α on time-consuming and accuracy

	10 degrees	20 degrees	30 degrees	40 degrees	50 degrees	60 degrees	70 degrees	80 degrees
time-consuming (in seconds)	194.7	192.4	191.5	190.9	189.9	187.0	184.8	**183.1**
absolute trajectory error (in meters)	0.112	0.104	0.082	0.073	**0.069**	0.079	0.095	0.113

[1] See more details for on https://github.com/MichaelGrupp/evo.

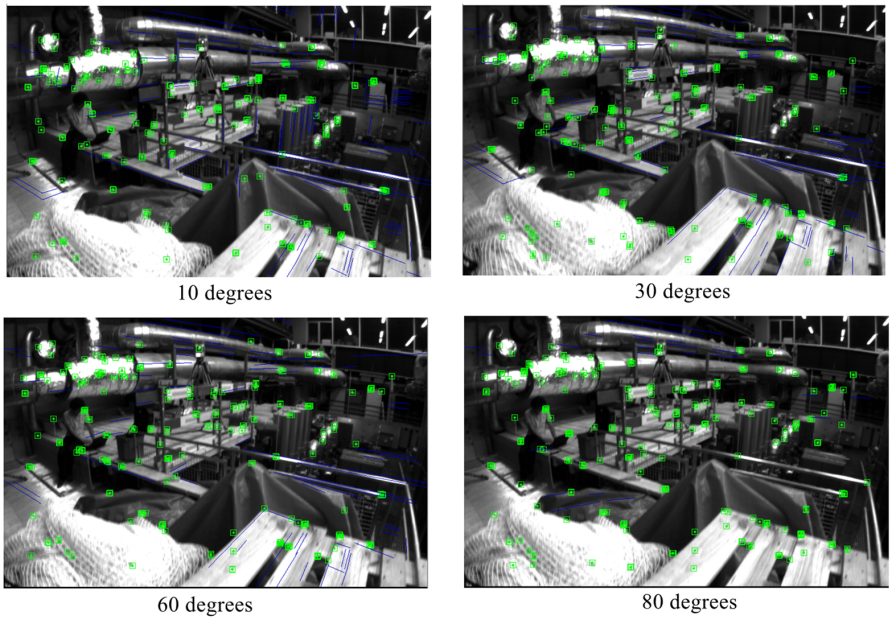

Figure 11. Structural line feature selection with various α.The larger the threshold α, the fewer structural lines (blue lines in the figure) are selected.

4.4 Experiment 2: Comparison experiment of the methods in this paper

In this experiment, all the above four methods (*our*, *ORB-SLAM2*, *NomalLine*, *ParallelLine*) are tested to complete the SLAM task on each of the five sequences (*MH01*, *MH05*, *V203*, *fr1_desk*, *fr3_str_tex_near*) introduced earlier, and the evaluation metrics of time-consuming and accuracy of each method for each sequence are discussed. We also repeated each test 5 times to obtain the final time-consuming and accuracy. In this experiment, we set the angle threshold α for *our* and *ParallelLine* methods to 50 degrees.

The results of this experiment are listed in Tab. 2 and Tab. 3. Taking the *V203* sequence as an example, the absolute trajectory error curves with running time for the four methods are plotted in the same graph as shown in Fig. 12, the absolute trajectory error distribution obtained through statistics is shown in Fig. 13.

Table 2. Time-consuming of the four methods (in seconds).

	our	*ORB-SLAM2*	*NomalLine*	*ParallelLine*
MH01	467.9	**234.1**	350.4	470.9
MH05	205.5	**145.1**	188.1	253.7
V203	189.9	**142.1**	147.9	195.4
fr1_desk	63.0	**29.7**	51.9	63.8
fr3_str_tex_near	114.8	**53.0**	91.6	109.1

Table 3. The absolute trajectory error of the four methods (in meters).

	our	*ORB-SLAM2*	*NomalLine*	*ParallelLine*
MH01	**0.045**	**0.045**	**0.045**	0.046
MH05	**0.050**	0.053	0.059	0.060
V203	**0.069**	0.117	0.189	0.103
fr1_desk	**0.014**	0.017	0.020	0.016
fr3_str_tex_near	**0.012**	0.015	0.014	0.013

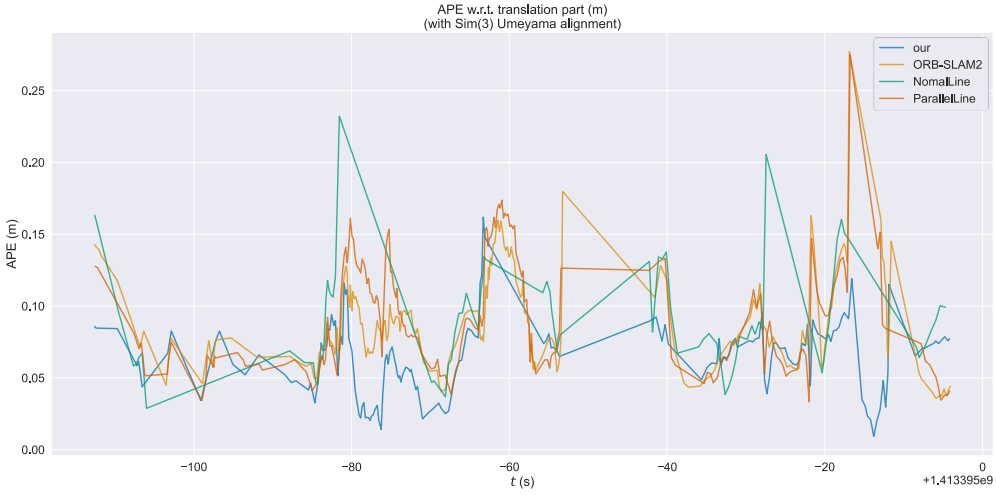

Figure 12. The plot of absolute trajectory error with time.

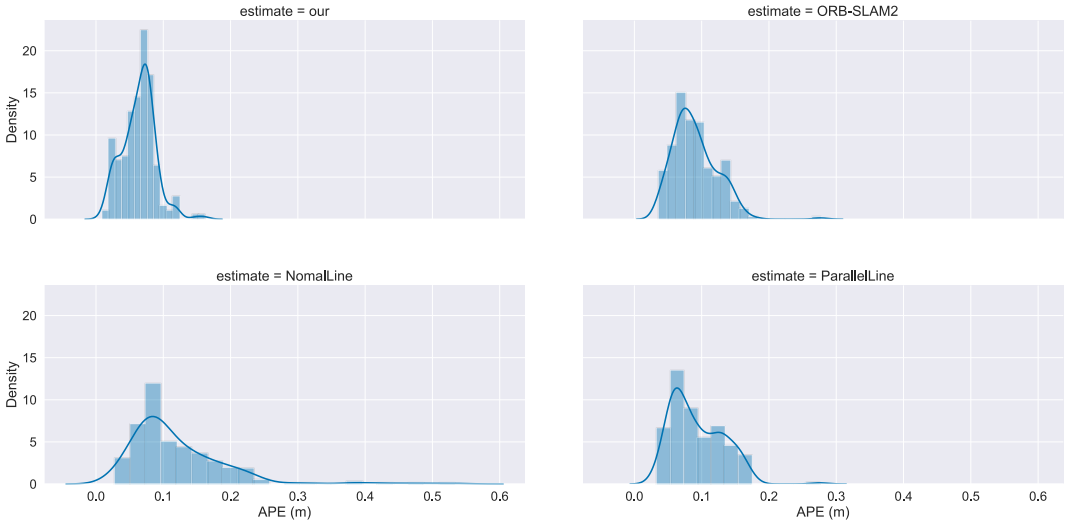

Figure 13. Absolute trajectory error distribution of the four methods.

In Tab. 2, we can find that for each sequence, the general trend of time-consuming of the four methods is that the original *ORB-SLAM2* is always the fastest, followed by *NomalLine*, and *our, ParallelLine* are the slowest. From quantitative comparison, we can know that although the time-consuming of *our is* increased by about 40% to 110% compared to the original *ORB-SLAM2*, the real-time performance is in fact guaranteed in our testing experiment. More specifically, we find that the increased time is mainly spent in three parts: line feature matching, vanishing point detection and structural line extraction, and the structural line feature selection procedure proposed in this paper has almost no effect on the time-consuming. Therefore, the point-line fusion Visual SLAM method with structural line features is always slower than the Visual SLAM method with just point features, the extra observation will inevitably bring extra computation. However, we would like to point out that there is no magnitude increasing of the time-consuming when integrating with line features, and all of them can basically run in real-time.

In terms of accuracy, as can be seen in Tab. 3, Fig. 12 and Fig. 13, the absolute trajectory error of *our* is the smallest on all experimental five sequences. Quantitatively, the accuracy of *our* is improved by about 15% to 40% compared with *ORB-SLAM2*, and about 15% compared with *NomalLine* (in some cases which have unstable matching of traditional line features, the larger improvement is expected). The accuracy of *ParallelLine* is close to that of *ORB-SLAM2*, this is probably because *ParallelLine* uses structural lines that are almost parallel to the direction of motion, as can be seen from the previous analysis know that these structural lines contribute less to the pose estimation, so the optimization process of point-line fusion at the back-end will almost degrade to the optimization of point features. Moreover, comparing the accuracy between *NomalLine* and *ORB-SLAM2*, it can be found that the method corresponding to *NomalLine* is unstable, and although the method introduces additional line features as observations, it is possible that the accuracy will be lower instead due to the various limitations imposed by the traditional line features, but this is hardly the case for the structural line features used in this paper.

5. CONCLUSION

In this paper, we propose a structural line feature selection method for improving the performance of indoor Visual SLAM. By considering the influence of directions of line features on pose estimation, vanishing points are used to generate structural line features and reasonable structural line features are selected based the information camera motion direction and structural line feature direction. In particular, only the structural line features that have a greater impact on the pose estimation can participate in the back-end optimization, which improves the pose estimation accuracy of Visual SLAM in indoor scenes. The experiments on five sequences of EuRoC and TUM show that after the selection procedure of the proposed method, the accuracy improvement is about 15%-40% compared to the original ORB-SLAM2 and about 15% compared to the PL-SLAM with only traditional line features. In terms of time efficiency, approximately extra 110% time is needed compared to the original ORB-SLAM2, but still the real-time requirement can be basically satisfied. In addition, according to the test dataset, the angle threshold value of proposed selection method is comprehensively explored and the best value of 50 degrees is advocated.

Only the monocular camera is considered in this work, and the uncertainty of its scale might cause some limitations in practical applications. In the future, we first want to further figure out the time-consuming issue and try to improve the computation speed, and then investigate the idea of fusion the structural lines with multi-sensor data, such as stereo cameras, depth camera etc.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foun dation of China (Grant No. 61871295).

REFERENCES

Bartoli, A., Sturm, P., 2003. The 3D line motion matrix and alignment of line reconstructions. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. I-I, doi: 10.1109.

Bartoli, A., Sturm, P., 2005. Structure-From-Motion Using Lines: Representation, Triangulation and Bundle Adjustment. Computer Vision & Image Understanding, 100(3):416-441.

Chandraker, M., Lim, J., Kriegman D., 2009. Moving in stereo: Efficient structure and motion using lines. Proc of IEEE International Conference on Computer Vision, pp. 1741–1748.

Coughlan, J. M., Yuille, A. L., 1999. Manhattan world: Compass direction from a single image by Bayesian inference. Proc of IEEE International Conference on Computer Vision, 941-947.

Davison, A. J., Reid, I. D., Molton, N. D., Stasse, O., 2007. MonoSLAM: Real-time single camera SLAM. IEEE Transactions Pattern Anal, vol. 29, no. 6, pp. 1052–1067.

Forster, C., Pizzoli, M., Scaramuzza, D., 2014. SVO: Fast semi-direct monocular visual odometry. IEEE International Conference on Robotics and Automation (ICRA), pp. 15-22.

Gomez-Ojeda, R., Briales, J., Gonzalez-Jimenez, J., 2016. PL-SVO: Semi-direct Monocular Visual Odometry by combining points and line segments. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Gomez-Ojeda, R., Moreno, F. -A., Zuñiga-Noël, D., Scaramuzza, D., Gonzalez-Jimenez, J., 2019. PL-SLAM: A Stereo SLAM System Through the Combination of Points and Line Segments. IEEE Transactions on Robotics, pp. 734-746.

Grompone von Gioi, R., Jakubowicz, J., Morel, J. -M, Randall, G., 2010. 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.

Han, B. X., Lu, H. M., Yu, Q. H., Zhang, L. L., 2021. Vanishing point estimation based on non-linear optimization in Manhattan world environments. Journal of Image and Graphics, 26(12):2931-2940.

Harris, C. G., StePhen, M., 1988. A Combined Corner and Edge Detector. Proceedings of the 4th Alvey Vision Conference, Manchester, 147-151.

Klein, G., Murray, D., 2007. Parallel tracking and mapping for small AR workspaces. Proc. IEEE ACM Proc. Int. Symp. Mixed Augmented Reality, pp. 225–234.

Lim, H., Jeon, J., Myung, H., 2022. UV-SLAM: Unconstrained Line-Based SLAM Using Vanishing Points for Structural Mapping. IEEE Robotics and Automation Letters, pp. 1518-1525.

Lu, X., Yaoy, J., Li, H., Liu, Y., Zhang, X., 2017. 2-Line Exhaustive Searching for Real-Time Vanishing Point Estimation in Manhattan World. IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 345-353.

Lu, Y., Song, D., 2015. Robust RGB-D odometry using point and line features. Proc of IEEE International Conference on Computer Vision, 3934-3942.

Ma, J., Wang, X., He, Y., Mei, X., Zhao, J., 2019. Line-Based Stereo SLAM by Junction Matching and Vanishing Point Alignment. IEEE Access, pp. 181800-181811.

Pumarola, A., Vakhitov, A., Agudo, A., 2017. PL-SLAM: real-time monocular visual SLAM with points and lines. Proc of IEEE International Conference on Robotics and Automation, 4503-4508.

Rublee, E., Rabaud, V., Konolige, K., Bradski, G., 2011. ORB: An efficient alternative to SIFT or SURF. 2011 International Conference on Computer Vision, pp. 2564-2571.

Sola, J., Vidal-Calleja, T., Devy, M., 2009. Undelayed initialization of linesegments in monocular SLAM. Proc. IEEE/RSJ Intell. Robots Syst., pp. 1553–1558.

Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W., 1999. Bundle Adjustment — A Modern Synthesis. Theory & Practice Springer-Verlag, vol 1883.

Zhang, L., Koch, R., 2013. 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.

Zhou, H., Zou, D., Pei, L., et al., 2015. StructSLAM: Visual SLAM With Building Structure Lines. IEEE Transactions on Vehicular Technology, vol. 64, no. 4, pp. 1364-1375.