IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 10, 2019, accepted December 13, 2019, date of publication December 17, 2019,
date of current version December 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2960282

Line-Based Stereo SLAM by Junction Matching and Vanishing Point Alignment

JIAYI MA[1,2], XINYA WANG[1], YIJIA HE[3], XIAOGUANG MEI[1], AND JI ZHAO[4]

[1]Electronic Information School, Wuhan University, Wuhan 430072, China
[2]Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems, Wuhan 430074, China
[3]Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[4]ReadSense Ltd., Shanghai 200000, China

Corresponding author: Ji Zhao (zhaoji84@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61773295 and Grant 61903279, and in part by the 111 Project under Grant B17040.

ABSTRACT In this paper, we propose a stereo simultaneous localization and mapping (SLAM) method based on line segments. For the front-end module of SLAM, we designed a novel method based on the coplanar junction detection, description, and matching. Then the junctions along with their multi-scale rotated BRIEF descriptors are used in other SLAM modules, including line tracking, mapping, and loop closure. The line extraction and matching thread runs at 20 $\sim$ 40Hz for stereo image sequences on a laptop, making it a practical front-end for line-based SLAM system. For the back-end module, a cost function is designed to minimize both of the reprojection error of line segments and alignment error of the vanishing points. The experimental results demonstrate that the proposed method exhibits more accurate localization and reconstruction than state-of-the-art line-based SLAM systems in line-rich environments.

INDEX TERMS Simultaneous localization and mapping (SLAM), line detection, vanishing point, visual odometry, loop closure.

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) has drawn much attention in recent years due to their broad applications. Currently, there are mainly two mainstreams of visual SLAM approaches including feature-based methods [1] and direct methods [2]. Feature-based methods consist of feature extraction and matching between frames. Then feature points' coordinates and camera poses are optimized by minimizing re-projection geometric error. Direct methods use raw pixel intensity for mapping and pose estimation by minimizing photometric errors. Usually feature methods are robust to illumination changes and geometric errors. Direct methods can create semi-dense maps which will benefit many applications.

Most of the feature-based SLAM methods utilize point feature for pose estimation and mapping. However, line segments are important features apart from point features especially in human-made environments, including both indoor environment and the so-called Manhattan world outdoors [3]. For such environments, consistent line segment maps have a high geometric expressiveness with respect to the underlying scene geometry. Besides, lines are more robust to illumination changes and have the potential for building semi-dense maps and recovering planes. As a result, line features may be an alternative to point features especially in untextured environments, when there are insufficient reliable feature points that can be detected. There are some works that utilize line segments for visual SLAM and visual odometry, such as [4]–[10].

Despite the progress of line-based SLAM, it is less mature compared with point-based SLAM. The potential reasons lie in several problems when introducing line segments in SLAM. First, current line segment detectors have many limitations for detection, tracking, and matching. To name a few, the detected line segments have a low repeatability rate across different images. Besides, the endpoints of line segments are unreliable and a long line segment may be divided into several short ones. Second, state-of-the-art line segment detection and matching methods are time-consuming. For example, the LSD [11] widely adopted in line-based SLAM takes 40 $\sim$ 50ms to process a 640 $\times$ 480 image, which makes

The associate editor coordinating the review of this manuscript and approving it for publication was Chenguang Yang.

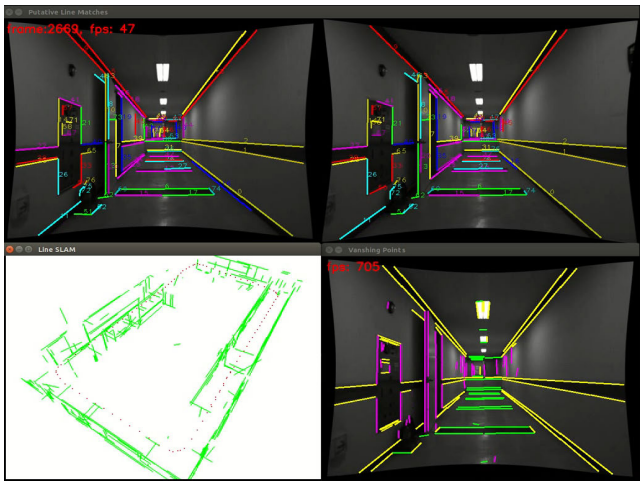


FIGURE 1. Typical results of our proposed SLAM method. Top Left and Top Right: line extraction and matching for a stereo image pair. Lines with the same color and number indicate a correspondence. Bottom Right: vanishing point extraction results. Lines with the same color share a common vanishing point. bottom Left: Line reconstruction and localization results.

line-based SLAM methods are difficult to be real-time. Third, compared to feature points, line segments are more complex for representation [12]. A 3D line has 4 degrees-of-freedom, and a line segment additionally has two endpoints for parameterization. When involving line segments in SLAM, the non-linearity and compactness of parameterization affect the performance.

In this paper, we make several improvements for line-based SLAM. For 640 × 480 stereo sequences, the line extraction and matching in our method is real-time by only one thread on a laptop. To the best of our knowledge, this is the first method in the literature that can achieve such efficiency, which will make the front-end in line-based SLAM towards practical. We also propose a cost function to exploit vanishing point (VP) alignment across frames to improve the accuracy of line-based SLAM. Fig. 1 demonstrates typical results of our proposed method. In summary, the main contributions include:

- We propose a novel method for junction and line matching. Specifically, we use multi-scale rotated BRIEF descriptors to construct line junction descriptors. The resulted line extraction and matching are much more efficient and accurate than state-of-the-art methods. Besides, we make a comprehensive evaluation of line segment detectors and recommend the Douglas-Peucker algorithm for line-based SLAM.
- A cost function is proposed for vanishing point alignment in back-end optimization. The cost function can be seamlessly integrated into the optimization framework of line-based SLAM, such as SLSLAM [6] and PL-VIO [13].
- For line-based SLAM, we propose a novel loop closure method that is based on junctions of lines. It uses the bag-of-words representation of junction descriptors to detect loops. The junction descriptors are byproducts of the line segment matching method, and it does not take additional time to extract.

We call our method *JunctionSLAM*, since most of the modules in our methods rely on coplanar junctions.

In the following text, first we introduce the line detection and matching method in Section III, which is the most important module in front-end of our SLAM method. Section II introduces the related work. In Section IV, we introduce the observation models and loss functions for line features, which are used to design the back-end of our SLAM method. Next, the whole SLAM method is briefly summarized in Section V. Finally, the experimental results are presented in Section VI, and the conclusions are drawn in Section VII.

II. RELATED WORK

SLAM has been widely applied in computer vision and robotics [1], [6], [8], [14], [15]. In this section, we briefly review the background material that our work is based on.

A. LINE SEGMENT BASED SLAM

Line segments have been integrated in filtering framework of SLAM [4], [16], [17]. Recently, line segments have been used in the optimization framework with bundle adjustment for stereo cameras [6], [10], [18] and monocular cameras [8]. There are also some works integrate line segments in point-based SLAM frameworks [7], [8], [10], [19]. Moreover, some works used the structural parallel lines as a constraint to estimate camera rotation. A group of parallel lines project to image plane will converge to a vanishing point (VP). For example, Camposeco *et al.* [20] deal VP as a measurement within an EKF-based visual-inertial odometry (VIO) system to improve the localization accuracy. Reference [21] use VP as a high-level landmark in a multilayer feature graph to directly calculate line landmarks direction in 3D space. Line segments are also used for implementing visual odometry [5], [22], [23].

Line segments have also been used in loop closure in visual SLAM. Being similar to loop closure with point features, typically discriminative line descriptors and bag-of-words representation [24] were adopted to detect loop closures in large-scale scenes [25]. Zhang *et al.* [26] proposed a vanishing point-based loop closure method in a line-based SLAM system.

B. FEATURE EXTRACTION, DESCRIPTION AND MATCHING

Point and Line segment extraction and matching is a long-standing problem in computer vision and robotics [27], [28]. Still, it is far from being solved. The most popular point feature extraction and description method in the SLAM area is ORB [29]. Popular line segment detector including Hough transform, Line Segment Detector (LSD) [11], EDLines [30], Fast Line Detector (FLD) [25], etc.

A few line descriptors have been proposed to describe the line segments. Most of them build gradient histograms around line segments, which are similar to point descriptors.

Representative line descriptor is Line Band Descriptor (LBD) [31]. Some works build descriptors for the junction of putative coplanar lines, such as warped regions [32], line intersection context feature [33] and Line-Junction-Line structures [34], [35].

Once the descriptors have been achieved, putative matches can be set up according to the similarity of descriptors. However, the presence of outliers in correspondence is inevitable due to ambiguities in the points' local appearance. There are many mismatch removal method for point correspondence [36]–[40].

C. GEOMETRY FOR LINE SEGMENTS AND VANISHING POINTS

The most natural way for line parameterization is using *Plücker* coordinates [41]. However, *Plücker* coordinates take 6 parameters to parameterize a line. Orthonormal representation allows minimum 4 parameters with an unconstrained optimization solver [12]. It is the most compact and has been successfully employed optimization framework [6].

Camera pose estimation from features is the core task in SLAM. Given line matches, typically 13 line correspondences across 3 frames are used to estimate the relative pose by trifocal tensor [41]. Minimal solver for recovering camera motion across two views of a calibrated stereo rig is studied in [42]. The algorithm can handle any assorted combination of point and line features across these 4 images.

Vanishing point (VP) is the intersection of several line segments in the image plane projected by a group of parallel lines. The geometric properties of VP are useful in many applications. The camera intrinsic parameters can be estimated by exploiting VPs [43]. Two VPs of different groups of parallel lines can be used to estimate camera rotation [44]. VP has also been used for relative pose estimation [45] and as a measurement within an EKF-based visual-inertial odometry (VIO) system [20]. In [21], VPs are used as high-level landmarks in a multilayer feature graph, and the direction of such landmarks are represented by corresponding VPs.

III. JUNCTION & LINE DETECTION AND MATCHING

Many of current line-based SLAM methods use LSD [11] to detect line segments, then they use LBD descriptor [31] for line segment matching [7], [8], [10], [18]. However, in the front-end of our method, we use different methods for line detection and matching. In this section, several improvements to state-of-the-art junction detection and junction/line matching will be introduced.

A. PUTATIVE COPLANAR JUNCTION DETECTION

A practical line-based SLAM should have an efficient and effective line segment detector. In section VI-A, we make a comprehensive evaluation of state-of-the-art line segment detectors. According to our evaluation, we adopt the Douglas-Peucker algorithm [46] to detect line segments due to its superior performance and high efficiency.

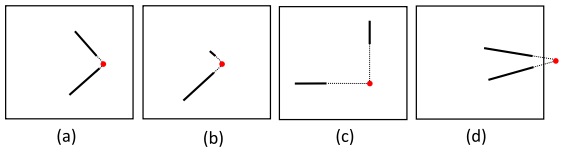


FIGURE 2. Heuristics to determine whether two line segments construct a valid junction. Solid lines are detected line segments, and red dots are their intersections. (a) a valid junction; (b) invalid since one of the line segments is too short; (c) invalid since the intersection is far away from line segments; (d) invalid since the intersection is out of the image plane.

Once the line segments are detected, putative coplanar junctions can be constructed. Usually, an image contains dozens to hundreds of line segments. Theoretically, any two line segments may construct a junction, so the potential number of junctions is large. To preserve meaningful junctions only, we use certain heuristics and carefully implement pruning strategies that are similar to those in previous literature [33], [35], [47].

The heuristics are explained in Fig. 2. Specifically, a putative coplanar ray-point-ray (RPR) junction should satisfy all of the following 3 conditions: (i) An RPR junction consists of two line segments, and their intersection is inside of the image plane. The two line segments are not necessarily intersected directly and may intersect in their extension. (ii) The length of both line segments should be above a threshold. (iii) The distance between the intersection and its line segments should be below a threshold [33]. After applying these 3 heuristics, an image usually contains $100 \sim 1,500$ junctions.

We use a triplet $\mathcal{J}(\mathbf{x}, \theta, \varphi)$ to represent the image of an RPR junction. Here, $\mathbf{x} \in \mathbb{R}^3$ is the junction's homogeneous coordinates in the normalized image. θ and φ are angles of the rays that construct this junction. For the convenience of subsequent junction matching, θ and φ are selected since their clockwise angle is between $(0, \pi)$.

B. JUNCTION DESCRIPTION AND MATCHING

A simple and efficient method is proposed for junction matching based on a multi-scale rotated BRIEF descriptor [48]. To build the rotated BRIEF descriptors for junctions, we need to determine the feature position, orientation, and scale of the junctions at first. As shown in Fig. 3, the junction position is the intersection of two line segments in the image plane. The orientation is set as the angle bisector of the junction in the image plane. The scale of the junction is difficult to determine. A simple strategy is using multiple scales. In this paper, 3 scales are used for each junction, including 10, 15, and 20 pixels. Once the position, orientation and scale of junctions are given, we extract their multi-scale BRIEF descriptors [48] as the junction description.

To determine the junction matching between two images, the efficient Hamming distance for binary strings is adopted. The matches inevitably have mismatches. A scheme for building putative matches and rejecting mismatches should be developed for junction matching.

In this paper, we use stereo rigs for SLAM. For rectified stereo image pairs, all epipolar lines are parallel in the

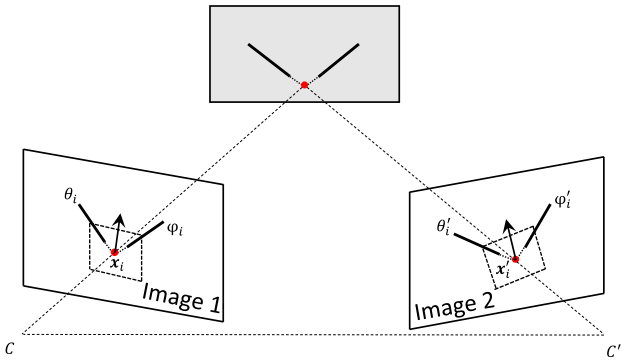


FIGURE 3. The junctions and their rotated BRIEF support regions. In this scene, there are 2 coplanar line segments in 3D space. In two images, solid lines are detected line segments. Arrows are the angle bisectors of the detected junctions. Squares are the support regions of rotated BRIEF descriptors.

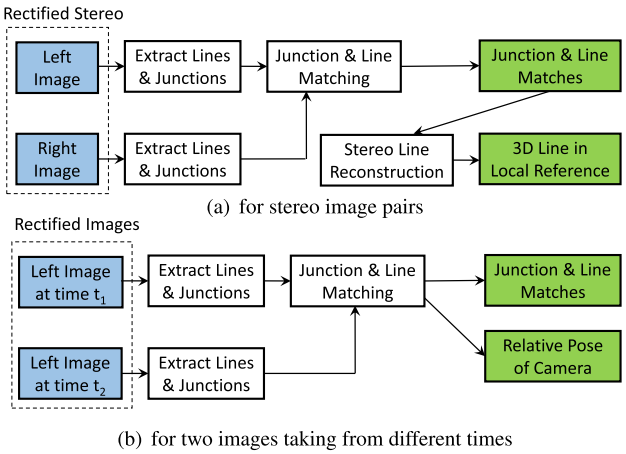


FIGURE 4. Flowcharts for line extraction and matching, and reconstruction. They are building blocks for the front-end of our proposed SLAM method. Blue blocks are inputs, and green blocks are outputs.

rectified image planes. Given a junction in the left image, we search its potential matches in the right image along the epipolar line. If there exist junctions near the corresponded epipolar line, the Hamming distance is used to verify whether they could construct a match. The procedure is shown in Fig. 4(a).

For two images that are from different times or from arbitrary viewpoints, the epipolar lines are unknown in advance. We use the Hamming distance to determine putative junction matching. Given a junction in one image, we find its closest neighbor and the second-closest neighbor according to Hamming distance. Taking the ratio of distance from the closest neighbor to the distance of the second closest, they are accepted as a match if this ratio is below a threshold.

Due to the ambiguities of local appearance, the junction matches inevitably have outliers. Denote a pair of matched junctions from image I_1 and I_2 as $\mathcal{J}_i^{I_1}(\mathbf{x}_i, \theta_i, \varphi_i)$ and $\mathcal{J}_i^{I_2}(\mathbf{x}'_i, \theta'_i, \varphi'_i)$. If these two junctions are constructed by the same two coplanar lines in 3D space, the intersections can be viewed as feature points and they satisfy the epipolar geometry

$$\mathbf{x}_i'^{\top}\mathbf{E}\mathbf{x}_i = 0, \tag{1}$$

where $\mathbf{E}$ is the essential matrix. The standard 5 points with RANSAC can be used to remove outliers and estimate essential matrix $\mathbf{E}$. Sampson error [41] is adopted to determine whether a junction match $(\mathcal{J}_i^{I_1}, \mathcal{J}_i^{I_2})$ is inlier. $(\mathcal{J}_i^{I_1}, \mathcal{J}_i^{I_2})$ is an inlier

$$d_{\text{point}}^{(i)} = \frac{(\mathbf{x}_i'^{\top}\mathbf{E}\mathbf{x}_i)^2}{(\mathbf{E}\mathbf{x}_i)_1^2 + (\mathbf{E}\mathbf{x}_i)_2^2 + (\mathbf{E}^{\top}\mathbf{x}'_i)_1^2 + (\mathbf{E}^{\top}\mathbf{x}'_i)_2^2}. \tag{2}$$

Once the junction matching has been finished, rotation and translation between these two images can be extracted from essential matrix $\mathbf{E}$ if they are needed. The procedure is summarized in Fig. 4(b).

C. FROM JUNCTION MATCHING TO LINE MATCHING

After a junction match across two images has been obtained, we can extract 2 line matches. The following observation can help us solve the ambiguity for these 2 line matches.

Since θ and φ are selected such that their clockwise angle in image plane is between $(0, \pi)$, θ_i always corresponds to θ'_i and φ_i corresponds to φ'_i.

IV. OBSERVATION MODELS AND LOSS FUNCTIONS FOR LINE FEATURES

In our method, line-based bundle adjustment is used to optimize the camera pose and line coordinates in 3D space. We exploit two types of observations in the image plane for 3D lines: line segments and vanishing points for parallel lines. In this section, first we introduce the observation models for these two observations. Then we construct a cost function that considers the reprojection errors for both line segments and vanishing points.

A. OBSERVATION MODEL FOR LINE SEGMENTS

A line segment in an image plane can be represented by two endpoints, $\mathbf{x}_s = (x_s, y_s, 1)^{\top}$ and $\mathbf{x}_e = (x_e, y_e, 1)^{\top}$. To build up the relationship between 3D line and 2D line segment in an image plane, we need to transform the 3D line in the world reference to the camera reference and then project it to the image plane. We adopt two parameterizations for a 3D line like that in [6]. Plücker line coordinates are used for transformation and projection due to its simplicity. Orthonormal representation is used for optimization due to its compactness.

A 3D line $\mathcal{L}$ in Plücker coordinate is represented by $\mathcal{L} = (\mathbf{n}, \mathbf{d})^{\top} \in \mathbb{R}^6$, where $\mathbf{d} \in \mathbb{R}^3$ is the line direction vector, and $\mathbf{n} \in \mathbb{R}^3$ is the normal vector to the plane determined by the line and the coordinate origin. A 3D rigid body motion $\mathbf{T} \in SE(3)$ is defined by $\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$, where $\mathbf{R} \in SO(3)$ is a 3×3 rotation matrix and $\mathbf{t} = (t_x, t_y, t_z)^{\top} \in \mathbb{R}^3$ is a translation vector in 3D space.

Given the transformation matrix

$$\mathbf{T}_{cw} = \begin{bmatrix} \mathbf{R}_{cw} & \mathbf{t}_{cw} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{3}$$

from the world frame W to the camera frame C, we can transform the Plücker representation of a line by [49]

$$\mathcal{L}_c = \begin{bmatrix} \mathbf{n}_c \\ \mathbf{d}_c \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{cw} & [\mathbf{t}_{cw}]_\times \mathbf{R}_{cw} \\ \mathbf{0} & \mathbf{R}_{cw} \end{bmatrix} \mathcal{L}_w, \tag{4}$$

where $[\cdot]_\times$ is the skew-symmetric matrix of a vector, and subscripts c and w represent camera and world respectively. After representing a line in camera frame, we can project it to the camera image plane by [6]

$$\mathbf{l} = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} = \mathcal{K}\mathbf{n}_c = \begin{bmatrix} f_y & 0 & 0 \\ 0 & f_x & 0 \\ -f_y c_x & -f_x c_y & f_x f_y \end{bmatrix} \mathbf{n}_c, \tag{5}$$

where $\mathcal{K}$ is the projection matrix of line feature. When projecting a line to the normalized image plane, $\mathcal{K}$ is a identity matrix. From projection equation (5), the coordinate of a line segment projected by 3D line is only related with the normal vector $\mathbf{n}$.

For point features, the reprojection error of a 3D point is the image distance between the projected point and the observed point. For line features, the reprojection error can be defined as the distance from two endpoints of a line segment to the projected line. Formally, the reprojection error for a 3D line k in camera frame i is defined as

$$\mathbf{e}_l(i,k) = \begin{bmatrix} d(\mathbf{x}_s^{i,k}, \mathbf{l}^k) \\ d(\mathbf{x}_e^{i,k}, \mathbf{l}^k) \end{bmatrix} \tag{6}$$

with $d(\mathbf{x}, \mathbf{l})$ being the distance from point $\mathbf{x}$ to line $\mathbf{l}$

$$d(\mathbf{x}, \mathbf{l}) = \frac{\mathbf{x}^\top \mathbf{l}}{\sqrt{l_1^2 + l_2^2}}, \tag{7}$$

where $\mathbf{x}_s^{i,k}$ and $\mathbf{x}_e^{i,k}$ are the two endpoints for the projected line segment of $\mathbf{l}^k$.

B. OBSERVATION MODEL FOR VANISHING POINTS

A detected vanishing point in an image plane is represented by $\mathbf{v} = (v_x, v_y, 1)^\top$. Given the normal direction $\mathbf{d}_c$ of corresponding parallel lines in camera frame C, the vanishing point in image plane can be predicted by [41]

$$\mathbf{v}_p = \mathbf{K}\mathbf{d}_c = \mathbf{K}\mathbf{R}_{cw}\mathbf{d}_w, \tag{8}$$

where $\mathbf{K}$ is the camera intrinsic matrix, subscripts c and w represent camera and world respectively, and subscripts p means prediction.

A straightforward way to compute the reprojection error of a VP is to calculate the image distance between the observation $\mathbf{v}$ and the prediction $\mathbf{v}_p$ in the normalized image plane. However, as shown in Fig. 5, the error for VP in the normalized image plane is unbound and the error may change drastically when optimizing the line parameters. To remedy this problem, we define the error at a unit sphere centered on the camera's projection center. Denote $\mathbf{v}^{i,k}$ as the VP for line k in camera frame i, then the reprojection errors for this VP is defined as

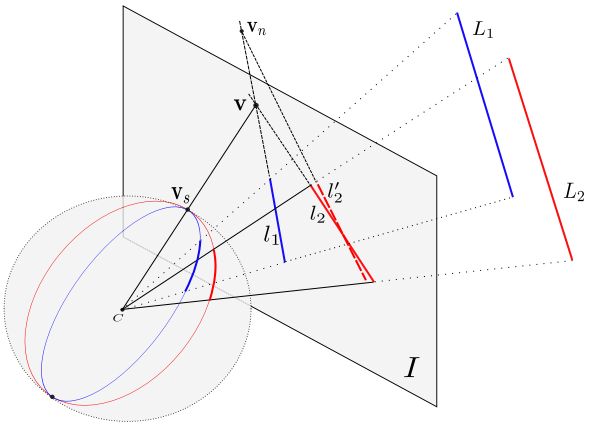


FIGURE 5. Parallel lines $\mathbf{L_1}$ and $\mathbf{L_2}$ in 3D space are projected to the image plane $\boldsymbol{I}$ as $\mathbf{l_1}$, $\mathbf{l_2}$. Point $\mathbf{v}$ is the groundtruth VP, and $\mathbf{v_n}$ is a predicted VP by $\mathbf{L_1}$ and a perturbation of $\mathbf{L_2}$. The reprojection error between $\mathbf{v}$ and $\mathbf{v_n}$ in image plane $\boldsymbol{I}$ is unbound and sensitive. We define the reprojection error by projecting vanishing points to a unit sphere.

$$\begin{aligned} \mathbf{e}_v(i,k) &= \frac{\mathbf{v}^{i,k}}{||\mathbf{v}^{i,k}||} - \frac{\mathbf{v}_p^{i,k}}{||\mathbf{v}_p^{i,k}||} \\ &= \frac{\mathbf{v}^{i,k}}{||\mathbf{v}^{i,k}||} - \frac{\mathbf{K}\mathbf{R}_{cw}^i \mathbf{d}_w^k}{||\mathbf{K}\mathbf{R}_{cw}^i \mathbf{d}_w^k||}. \end{aligned} \tag{9}$$

Since the unit sphere is a bound space, this error function can balance the reprojection error for all VPs in a fair way.

C. COST FUNCTION FOR BUNDLE ADJUSTMENT

To optimize the camera poses and line coordinates, a cost function is constructed by jointly consider the observation models for line segments and vanishing points

$$\mathcal{C} = \sum_{i,k} \rho(\mathbf{e}_l^\top(i,k)\Sigma_l^{-1}\mathbf{e}_l(i,k)) + \sum_{i,j} \rho(\mathbf{e}_v^\top(i,j)\Sigma_v^{-1}\mathbf{e}_v(i,j)), \tag{10}$$

where $\rho(\cdot)$ is the robust Cauchy cost function, and Σ_l^{-1} and Σ_v^{-1} are information matrices for line segments and vanishing points. In this paper, the information matrices are set as identity matrices. In the first term, k is the index for all lines. In the second term, j is the index for lines that belongs to a group of parallel lines.

V. STEREO VISUAL SLAM BASED ON LINE SEGMENTS AND VANISHING POINTS

A. SYSTEM OVERVIEW

Our proposed JunctionSLAM is based on graph optimization. Being similar to the mainstreams of feature-based SLAM systems, it has three threads: motion estimation, local mapping, and loop closing. The system overview is shown in Fig. 6. The details about each component will be described in the following text.

B. MOTION ESTIMATION

For each new stereo frame, we use our line detection and matching method as described in section III to build the line

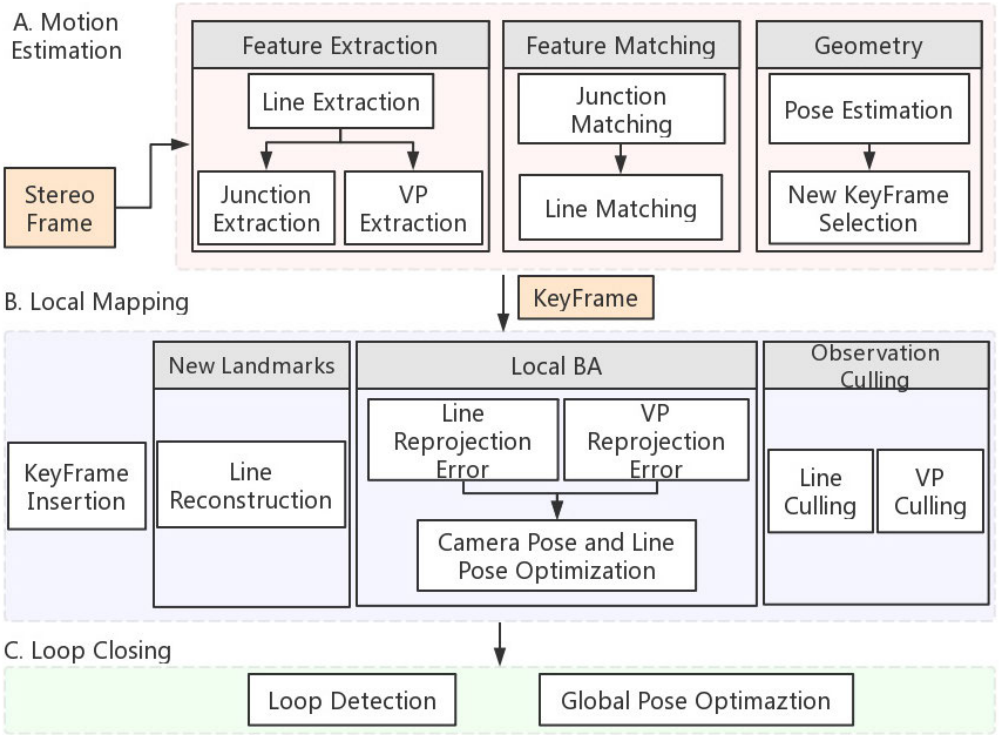


FIGURE 6. JunctionSLAM system overview, showing main steps performed by the tracking, local mapping and loop closing threads.

matches. We also use the VP extraction method in [50] to cluster lines in the left image and compute the coordinates of VPs in the image plane. The line segments and VPs of each frame will be served as observations.

We use the method in [42] to estimate motions by trifocal tensor geometry. For a stereo rig, the minimal solver for motion estimation needs 3 line matches between the left image of the current frame and the latest stereo keyframe. We use 3 line matches across 3 images: the left image of the lastest keyframe, and two images of the new stereo frame. This minimal solver together with the RANSAC framework is used to estimate relative pose. When the translation or rotation of relative pose between the new frame and the lastest keyframe exceeds threshold η_t or η_r, we will select the new frame as a keyframe.

C. LOCAL MAPPING

Once a new keyframe is inserted into the pose graph, the 3D coordinates of line segments in this new keyframe will be reconstructed as that in SLSLAM by intersecting two planes [6]. Then we will select N latest keyframes along with the new keyframe as an active frame. Camera poses and line segments belonging to active frames will be refined by local bundle adjustment. Finally, after poses are refined, a culling strategy will be used to remove outliers from line matches or VP clusters. Specifically, if the reprojection error of a line segment exceeds a threshold $\eta_l = 5$, this line segment will be removed. Similarly, if the reprojection error between a VP observation and the predicted VP exceeds a threshold of $\eta_v = 0.3$, this VP observation will be removed.

D. LOOP CLOSURE DETECTION

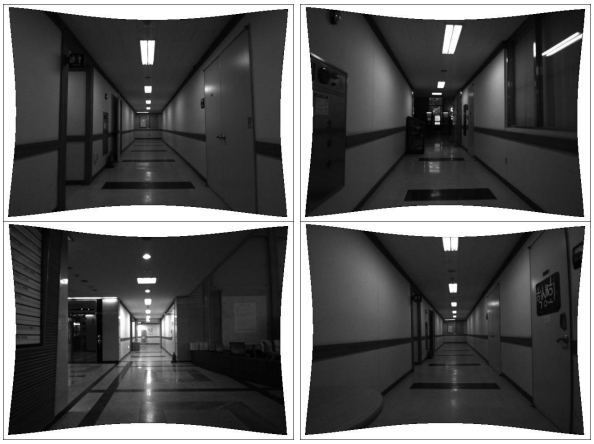
Our loop closure detection is similar to that in ORB-SLAM [1]. The difference lies in that we do not use oriented FAST to detect feature points like that in ORB feature detector [29]. Instead, we use the extracted junctions described in section III-A as feature points.

Our method also uses the bag-of-words feature representation to perform loop closure detection and relocalization. The visual words are organized by a hierarchical tree called a vocabulary tree [24]. We use the vocabulary tree provided by ORB-SLAM, which is built offline by clustering a large number of BRIEF descriptors extracted from an image dataset.

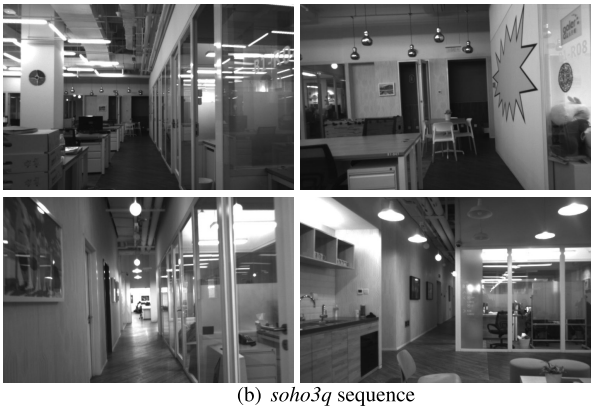
The loop closure thread compares current images to the previous keyframes $K_i, i = 1, 2, \ldots, t$. We query the keyframe dataset and discard all those keyframes whose similarity score is lower than a predefined threshold. To accept a loop candidate we must detect consecutively 3 loop candidates that are consistent. For these loop candidates, we further perform 5-point method with the RANSAC framework for geometric verification. If there are sufficient matches passed the geometric verification, a loop is detected.

VI. EXPERIMENTAL RESULTS

We use the stereo sequence *it3f* in SLSLAM [6] to test our method. It contains 5, 442 pairs of stereo images, and each image has a resolution of 640 × 480. We also captured a stereo sequence called *soho3q* by a stereo camera with global shutters. It contains 1, 640 stereo image pairs, and each image has a resolution of 752 × 480. The sequence is taken in a



(a) *it3f* sequence



(b) *soho3q* sequence

FIGURE 7. Sample images of the test sequences.

coworking space, as shown in Fig. 7. Compared with previous line SLAM sequences, it contains more line segments on average. It is challenging for SLAM since it contains image saturation caused by ceiling lights and reflection caused by glass walls.[1] All of the experiments are performed on a laptop with Intel i7-5500U CPU @ 2.40 GHz QuadCore.

A. LINE & JUNCTION DETECTION AND MATCHING

There are many line segment detectors in computer vision and robotics communities. However, there does not exist any comprehensive evaluation for these methods. We compare 4 state-of-the-art methods that have open-sourced implementation, including line segment detector (**LSD**) [11], **EDLine** [30], Fast Line Detector (**FLD**) [25], and Douglas-Peucker algorithm (**DP**) [46]. For LSD and FLD, the source codes were provided by the authors. These two methods also have been integrated into openCV. For EDLine, there is an open-sourced implementation that has been integrated into LBD [31].[2] We do not use the multi-scale processing to make it as the same as the original paper. For the Douglas-Peucker algorithm, we use the source code

[1]https://drive.google.com/open?id = 0B_tUbCEawNQlZThPeW1rNmZvbzA

[2]http://www.mip.informatik.uni-kiel.de/tiki-index.php?page = Lilian+Zhang

TABLE 1. Performance of line segment detection on *it3f*.

	LSD [11]	EDLine [30]	FLD [25]	DP [46]
time (ms/image)	34.0	14.6	18.1	**7.4**
avg line segment num	130.3	204.4	68.1	147.9
avg length (pixels)	48.2	81.5	80.9	46.0
near-duplicate lines	few	Many	few	few
avg junction number	238.9	741.0	104.5	310.5
avg junction matches	137.9	198.5	60.2	149.8
avg line matches	82.9	97.4	40.3	87.9

TABLE 2. Performance of line segment detection on *soho3q*.

	LSD [11]	EDLine [30]	FLD [25]	DP [46]
time (ms/image)	43.5	21.3	57.7	**11.5**
avg line segment num	420.7	534.0	199.1	475.8
avg length (pixels)	37.9	68.6	64.0	36.2
near-duplicate lines	few	Many	few	few
avg junction number	797.4	1626.3	224.2	1014.6
avg junction matches	345.9	381.3	100.0	380.1
avg line matches	223.4	208.6	84.3	238.6

provided by the authors of [5], which is part of the Line Vision Library.[3] All these 4 methods were implemented in C++ programming language. LSD and EDLine do not need any parameter tuning. In both FLD and DP, the minimal line length is set as 12 pixels. When performing the experiments in this subsection, only one CPU core is used.

For line segment detection, there is no sufficient evaluation criterion and benchmarks for performance comparison in literature. In this paper, 3 surrogate criteria are used, including runtime, line segment number, and the average length of line segments.

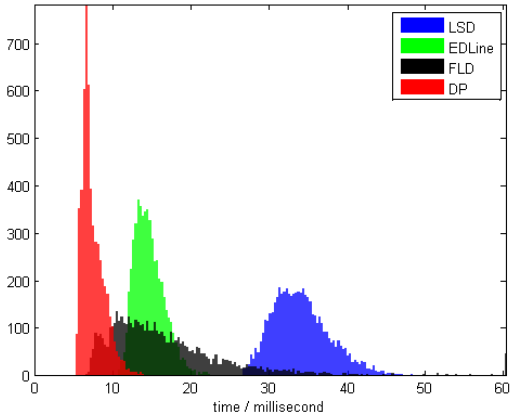
The quantitative results of line segment detection of *it3f* and *soho3q* are shown in Table 1 and Table 2, respectively. For efficiency consideration, DP is superior to other 3 methods. From line segment number, it is difficult to say which method is better, because this number is influenced by many factors, such as the recall of line segment and whether a long line segment is divided into short ones. From the average length of line segments, EDLine is the best and FLD is the second best. However, we observed that EDLine tends to produce many near-duplicate line segments. The histograms of runtime are shown in Fig. 8. It can be seen that the Douglas-Peucker algorithm has the smallest average runtime and the smallest standard deviation.

After the line segments are detected, we construct the junctions and perform junction and line matching by our method described in section III. Typical results of the line matching for stereo pairs are shown in Fig. 9. Table 1 and 2 also demonstrate our junction matching and line matching results based on different line segment detectors. Here three criteria are used, including junction number, matched

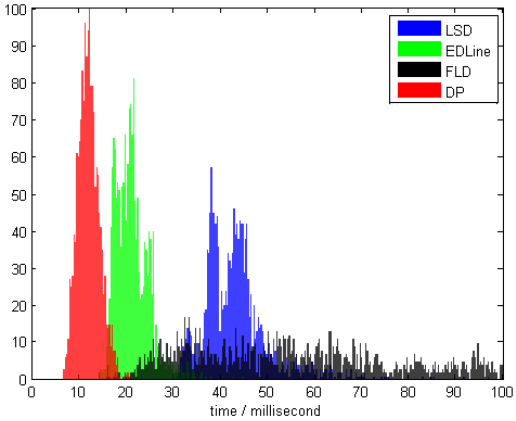
[3]https://bitbucket.org/lvl_dev/lvl

TABLE 3. Performance comparison for line description and matching on *it3f*.

	LBD description & matching [31]	our junction description & Matching
time (ms)	34.18	**8.45**
avg line matches	55.3	**82.9**

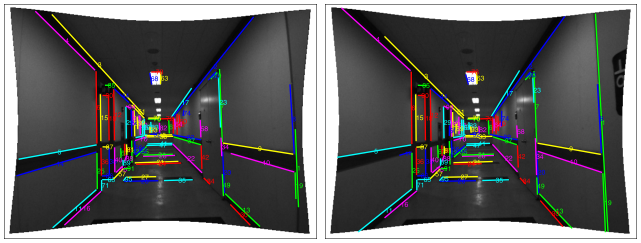


(a) for sequence *it3f*

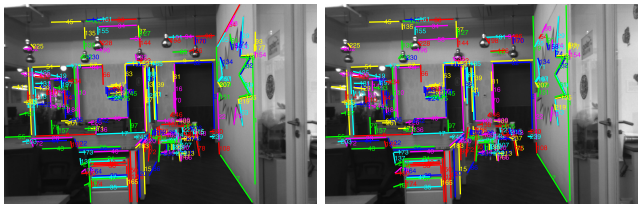


(b) for sequence *soho3q*

FIGURE 8. Runtime histogram for line segment detectors.



(a) for sequence *it3f*



(b) for sequence *soho3q*

FIGURE 9. The line matching results for a pair of stereo images. Matched line segments are characterized by the same color and the numbers in the middle of line segments (best view the electronic version of this article).

junction number, and matched line number in each image. Among these three criteria, the matched line number is the most important one. We can see EDLine and DP have the largest number of matched lines on *it3f* and *soho3q*, respectively.

We also compare our line matching method with the widely used LBD method [31]. For fairness consideration, the line detectors used in both methods are LSD provided by `openCV 3.1`. The results are shown in Table 3. Compared with LBD, our method is more efficient and has more matched lines. The potential reason is that LBD depends on the endpoints of detected line segments. While the detected endpoints are unstable, which may make the line matching inaccurate. In contrast, our method relies on junctions and does not depend on the endpoints of line segments.

After considering both the efficiency and quality, we recommend DP method for line segment detection in line-based SLAM, which is an alternative to popular LSD. If efficiency is not an issue, LSD would be our second recommendation, since it does not need parameter tuning and always produce reasonable results for all kinds of images. Combining the DP method and our junction & line matching method, we obtain a much more efficient line matching engine than stat-of-the-art methods.

B. SLAM RESULTS ON SYNTHETIC DATA

We construct a synthetic scene as shown in Fig. 10. We sampled 100 frames from a camera trajectory in 3D space which is composed of circular motion in X-Y plane and a sinusoidal motion along Z-axis. Each camera frame can observe a cube with 12 line segments. We have performed 50 times of experiments. In each experiment, Gaussian noise is added to the endpoints of observed line segments with zero-mean and a standard deviation of $\delta_{px} = 1$ pixel. As a result, the coordinates of 3 vanishing points are also degraded by this noise. When generating camera poses, the translation of camera poses is perturbed by a zero-mean Gaussian noise with standard deviation of $\delta_t = 0.1$ meter, and the rotation is perturbed by a zero-mean Gaussian noise with a standard deviation of $\delta_q = 5°$. We compared the line-based bundle adjustment method with and without VP constraints. After each run, we measure translation error and rotation error using the RMSE (root mean square error) of PRE (relative

TABLE 4. RMSE of relative pose error using different settings.

	BA with lines	BA with lines and VPs
translation error (unit: meter)	0.022	**0.013**
rotation error (unit: degree)	0.095	**0.042**

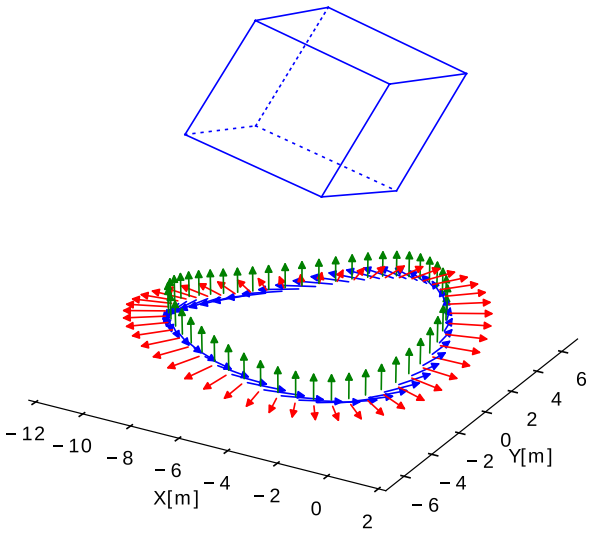
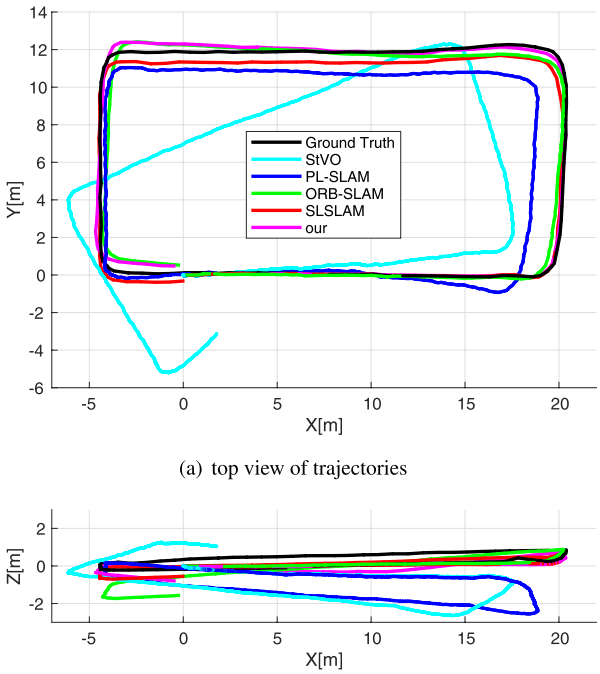


FIGURE 10. Synthetic scene. An upward-looking camera moves along a circular trajectory. A synthetic cube that is composed of 12 lines with three orthogonal directions is above the camera trajectory. A triplet of red, blue, and green arrows that share a common tail represents a camera pose.



(a) top view of trajectories

(b) front view of trajectories

FIGURE 11. Localization results comparison. Point (0, 0, 0) is the starting point of a trajectory. (unit: meter). (a) top view of trajectories. (b) front view of trajectories.

pose error) [51]. Table 4 shows the mean error of 50 experiments. It can be seen that the rotation error and translation error are reduced by involving VP alignment in the bundle adjustment.

C. SLAM RESULTS ON REAL DATA

We compare the proposed JunctionSLAM with **StVO** [23], **PL-SLAM** [10], **ORB-SLAM** [1] and **SLSLAM** [6]. StVO and PL-SLAM which are recently proposed VO and SLAM methods based on line features for stereo sequences. SLSLAM is the first line-based SLAM method using an optimization framework. It is open-sourced except for its line extraction, matching, and tracking modules. We can reproduce the results for *it3f* sequence since it provides the line tracking results, while it can not process any newly captured sequence. ORB-SLAM is a state-of-the-art point-based SLAM method. In this paper, vanishing points were extracted by the method in [50]. We find the results are satisfactory and efficient.

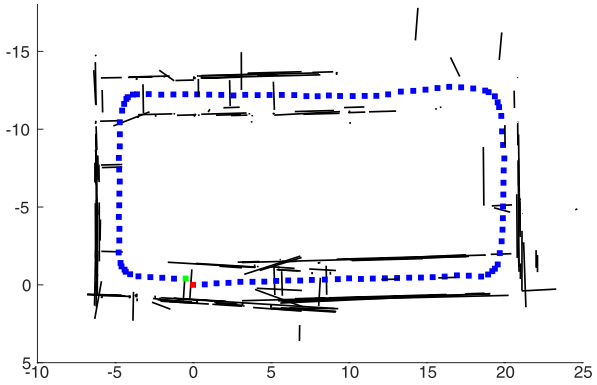
We run JunctionSLAM on sequence *it3f* which is widely used in line-based SLAM methods. For fair comparisons, we use the same parameters like that in SLSLAM (i.e., the keyframe selection thresholds are set as $\eta_t = 0.75$m and $\eta_r = 15°$). For other methods, the default parameters are used. The trajectories generated by our method, StVO, PL-SLAM, ORB-SLAM, and SLSLAM are shown in Fig. 11. Since the dataset does not provide the ground truth, the gap between the first pose and the last pose can be a metric for drift.

From Fig. 11, it can be seen that our proposed JunctionSLAM performs significantly better than StVO, PL-SLAM, and ORB-SLAM considering localization errors both in X-Y plane and Z-axis. Our method is slightly better than SLSLAM. Note that SLSLAM uses a GPU to make the line extraction and matching to be real-time. In contrast, our method can be faster than real-time even using a single thread of CPU. For ORB-SLAM, the localization error in Z-axis is as large as 1.65m, which means the point-based methods do not work well when there are few point features in the environment. Besides, we observed that there are significantly more keyframes in ORB-SLAM than that in line-based SLAM methods.

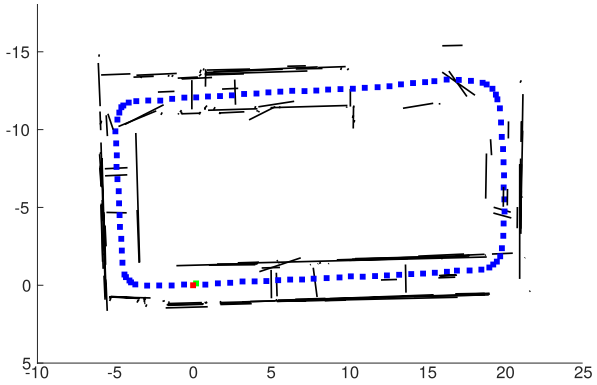
The top views of line reconstruction from the proposed JunctionSLAM system with or without loop closure are shown in Fig. 12. It can be seen that the loop closure can effectively reduce the localization drift. line reconstruction with loop closure is more consistent with the actual building structure.

TABLE 5. Processing time for *it3f* sequence (unit: millisecond).

line detection	junction description	line match	VP detection	pose estimate	bundle adjustment
9.43	3.07	2.31	1.70	2.55	25.84



(a) with VP alignment and without loop closure



(b) with VP alignment and loop closure

FIGURE 12. The line reconstruction of JunctionSLAM on *it3f* image sequence. Dotted lines are estimated trajectories, and each dot corresponds to a keyframe. The red dot and blue dot correspond to starting and ending points, respectively. (a) with VP alignment and without loop closure. (b) with VP alignment and loop closure.

JunctionSLAM runs in real-time for usual stereo sequences. Table 5 summarizes the processing time for the main components. It is worth to note that the line extraction and matching run at 20 $\sim$ 40Hz for usual stereo sequences on a laptop using a single thread, making it practical for line-based SLAM systems. The result for the whole stereo sequence is available from https://www.dropbox.com/s/4qehqxqkfr5r9qd/junctionSLAM_demo.avi

Despite its successful application to the real-world datasets, the proposed JunctionSLAM is suitable for line-rich environments only. For texture-rich environments, there might be insufficient line segment features, and point-feature-based SLAM systems are more suitable. Due to the complementary of point-based SLAM and line-based SLAM, it is promising that to combine these two features in the future work.

VII. CONCLUSION

In this paper, we propose a SLAM system based on coplanar junctions and vanishing points. Our contributions are threefold. First, by introducing junction matching and comprehensive evaluation of line segment detectors, we design a real-time line extraction and matching method. Second, a cost function is proposed that considers reprojection error of vanishing points. Third, a loop closure method based on junction descriptors is proposed. The effectiveness and efficiency of our method have been validated by real image sequences.

REFERENCES

[1] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[2] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2016.

[3] Y. Gao and A. L. Yuille, "Estimation of 3D category-specific object structure: Symmetry, manhattan and/or multiple images," *Int. J. Comput. Vis.*, vol. 127, no. 10, pp. 1501–1526, 2019.

[4] P. Smith, I. D. Reid, and A. J. Davison, "Real-time monocular SLAM with straight lines," in *Proc. Brit. Mach. Vis. Conf.*, 2006.

[5] J. Witt and U. Weltin, "Robust stereo visual odometry using iterative closest multiple lines," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 4164–4171.

[6] G. Zhang, J. H. Lee, J. Lim, and I. H. Suh, "Building a 3-D line-based map using stereo SLAM," *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1364–1377, Dec. 2015.

[7] S. Yang and S. Scherer, "Direct monocular odometry using points and lines," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2017, pp. 3871–3877.

[8] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "PL-SLAM: Real-time monocular visual SLAM with points and lines," in *Proc. Int. Conf. Robot. Autom.*, May 2017, pp. 4503–4508.

[9] H. Li, J. Yao, X. Lu, and J. Wu, "Combining points and lines for camera pose estimation and optimization in monocular visual odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2017, pp. 1289–1296.

[10] R. Gomez-Ojeda, F.-A. Moreno, D. Scaramuzza, and J. Gonzalez-Jimenez, "PL-SLAM: A stereo SLAM system through the combination of points and line segments," 2017, *arXiv:1705.09479*. [Online]. Available: https://arxiv.org/abs/1705.09479

[11] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722–732, Apr. 2010.

[12] A. Bartoli and P. Sturm, "Structure-from-motion using lines: Representation, triangulation, and bundle adjustment," *Comput. Vis. Image Understand.*, vol. 100, no. 3, pp. 416–441, 2005.

[13] Y. He, J. Zhao, Y. Guo, W. He, and K. Yuan, "PL-VIO: Tightly-coupled monocular visual–inertial odometry using point and line features," *Sensors*, vol. 18, no. 4, p. 1159, 2018.

[14] T. Liu, H. Liu, Y.-F. Li, Z. Chen, Z. Zhang, and S. Liu, "Flexible FTIR spectral imaging enhancement for industrial robot infrared vision sensing," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/TII.2019.2934728.

[15] T. Liu, Y.-F. Li, H. Liu, Z. Zhang, and S. Liu, "RISIR: Rapid infrared spectral imaging restoration model for industrial material detection in intelligent video systems," *IEEE Trans Ind. Informat.*, to be published, doi: 10.1109/TII.2019.2930463.

[16] E. Eade and T. Drummond, "Edge landmarks in monocular SLAM," *Image Vis. Comput.*, vol. 27, no. 5, pp. 588–596, 2009.

[17] H. Zhou, D. Zou, L. Pei, R. Ying, P. Liu, and W. Yu, "StructSLAM: Visual SLAM with building structure lines," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1364–1375, Apr. 2015.

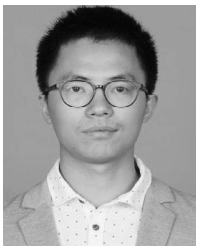
[18] X. Zuo, X. Xie, Y. Liu, and G. Huang, "Stereo visual SLAM with point and line features," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2017, pp. 1775–1782.
[19] R. Gomez-Ojeda, J. Briales, and J. Gonzalez-Jimenez, "PL-SVO: Semi-direct monocular visual odometry by combining points and line segments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2016, pp. 4211–4216.
[20] F. Camposeco and M. Pollefeys, "Using vanishing points to improve visual-inertial odometry," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2015, pp. 5219–5225.
[21] Y. Lu and D. Song, "Visual navigation using heterogeneous landmarks and unsupervised geometric constraints," *IEEE Trans. Robot.*, vol. 31, no. 3, pp. 736–749, Jun. 2015.
[22] M. Chandraker, J. Lim, and D. Kriegman, "Moving in stereo: Efficient structure and motion using lines," in *Proc. Int. Conf. Comput. Vis.*, 2009, pp. 1741–1748.
[23] R. Gomez-Ojeda and J. Gonzalez-Jimenez, "Robust stereo visual odometry through a probabilistic combination of points and line segments," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2016, pp. 2521–2526.
[24] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2006, pp. 2161–2168.
[25] 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 *Proc. IEEE Int. Conf. Robot. Autom.*, May 2014, pp. 5550–5557.
[26] G. Zhang, D. H. Kang, and I. H. Suh, "Loop closure through vanishing points in a line-based monocular SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 4565–4570.
[27] M. Chen, Z. Shao, C. Liu, and J. Liu, "Scale and rotation robust line-based matching for high resolution images," *Optik-Int. J. Light Electron Opt.*, vol. 124, no. 22, pp. 5318–5322, 2013.
[28] M. Chen and Z. Shao, "Robust affine-invariant line matching for high resolution remote sensing images," *Photogramm. Eng. Remote Sens.*, vol. 79, no. 8, pp. 753–760, 2013.
[29] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, 2011.
[30] C. Akinlar and C. Topal, "EDLines: A real-time line segment detector with a false detection control," *Pattern Recognit. Lett.*, vol. 32, no. 13, pp. 1633–1642, Oct. 2011.
[31] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency," *J. Vis. Commun. Image Represent.*, vol. 24, no. 7, pp. 794–805, 2013.
[32] E. Vincent and R. Laganiere, "Junction matching and fundamental matrix recovery in widely separated views," in *Proc. Brit. Mach. Vis. Conf.*, 2004.
[33] H. Kim and S. Lee, "A novel line matching method based on intersection context," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 1014–1021.
[34] K. Li, J. Yao, X. Lu, L. Li, and Z. Zhang, "Hierarchical line matching based on line–junction–line structure descriptor and local homography estimation," *Neurocomputing*, vol. 184, pp. 207–220, Apr. 2016.
[35] K. Li and J. Yao, "Line segment matching and reconstruction via exploiting coplanar cues," *ISPRS J. Photogram. Remote Sens.*, vol. 125, pp. 33–49, Mar. 2017.
[36] J. Ma, X. Jiang, J. Jiang, J. Zhao, and X. Guo, "LMR: Learning a two-class classifier for mismatch removal," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 4045–4059, Aug. 2019.
[37] J. Ma, J. Zhao, J. Jiang, H. Zhou, and X. Guo, "Locality preserving matching," *Int. J. Comput. Vis.*, vol. 127, no. 5, pp. 512–531, 2019.
[38] Z. Min, H. Ren, and M. Q.-H. Meng, "Statistical model of total target registration error in image-guided surgery," *IEEE Trans. Autom. Sci. Eng.*, to be published, doi: 10.1109/TASE.2019.2909646.
[39] Z. Min, J. Wang, and M. Q.-H. Meng, "Joint rigid registration of multiple generalized point sets with hybrid mixture models," *IEEE Trans. Autom. Sci. Eng.*, to be published, doi: 10.1109/TASE.2019.2906391.
[40] Z. Shao, M. Chen, and C. Liu, "Feature matching for illumination variation images," *J. Electron. Imag.*, vol. 24, no. 3, 2015, Art. no. 033011.
[41] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
[42] V. Pradeep and J. Lim, "Egomotion estimation using assorted features," *Int. J. Comput. Vis.*, vol. 98, no. 2, pp. 202–216, 2012.
[43] B. Caprile and V. Torre, "Using vanishing points for camera calibration," *Int. J. Comput. Vis.*, vol. 4, no. 2, pp. 127–139, Mar. 1990.
[44] Y. Salaün, R. Marlet, and P. Monasse, "Robust and accurate line- and/or point-based pose estimation without manhattan assumptions," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 801–818.
[45] J.-K. Lee and K.-J. Yoon, "Real-time joint estimation of camera orientation and vanishing points," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1866–1874.
[46] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica*, vol. 10, no. 2, pp. 112–122, 1973.
[47] J. Zhao, L. Kneip, Y. He, and J. Ma, "Minimal case relative pose computation using ray-point-ray features," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
[48] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "BRIEF: Computing a local binary descriptor very fast," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1281–1298, Nov. 2012.
[49] A. Bartoli and P. Sturm, "The 3D line motion matrix and alignment of line reconstructions," *Int. J. Comput. Vis.*, vol. 57, no. 3, pp. 159–178, 2004.
[50] M. Nieto and L. Salgado, "Real-time robust estimation of vanishing points through nonlinear optimization," *Proc. SPIE*, vol. 7724, May 2010, Art. no. 772402.
[51] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. Int. Conf. Intell. Robot Syst.*, Oct. 2012, pp. 573–580.

JIAYI MA received the B.S. degree in information and computing science and the Ph.D. degree in control science and engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2008 and 2014, respectively.

From 2012 to 2013, he was an Exchange Student with the Department of Statistics, University of California at Los Angeles, Los Angeles, CA, USA. He held a postdoctoral position with the Electronic Information School, Wuhan University, from August 2014 to November 2015, and received an accelerated promotion to Associate Professor and Full Professor, in December 2015 and December 2018, respectively. He has authored or coauthored more than 120 refereed journal articles and conference papers, including the IEEE TPAMI/TIP/TSP/TNNLS/TIE/TGRS/TCYB/TMM/TCSVT, IJCV, CVPR, ICCV, IJCAI, AAAI, ICRA, IROS, ACM MM, and so on. His research interests include computer vision, machine learning, and pattern recognition.

Dr. Ma has been identified in the 2019 Highly Cited Researchers list from the Web of Science Group. He was a recipient of the Natural Science Award of Hubei Province (First Class), the Chinese Association for Artificial Intelligence(CAAI) Excellent Doctoral Dissertation Award (a total of eight winners in China), and the Chinese Association of Automation(CAA) Excellent Doctoral Dissertation Award (a total of ten winners in China). He is an Editorial Board Member of *Information Fusion* and *Neurocomputing*, and a Guest Editor of *Remote Sensing*.

XINYA WANG received the B.S. degree from the Electronic Information School, Wuhan University, Wuhan, China, in 2018. She is currently pursuing the master's degree with the Multi-Spectral Vision Processing Laboratory, Wuhan University. Her current research interests include neural networks, machine learning, and image processing.



YIJIA HE received the B.E. degree in automatic control science and engineering from Hunan University, Changsha, China, in 2013, and the Ph.D. degree in control science and engineering from the Institute of Automation, Chinese Academy of Sciences, in 2018. His research interests include visual SLAM and sensor fusion for mobile robot localization.

XIAOGUANG MEI received the B.S. degree in communication engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2007, the M.S. degree in communications and information systems from Huazhong Normal University, Wuhan, in 2011, and the Ph.D. degree in circuits and systems from the HUST, in 2016. From 2010 to 2012, he was a Software Engineer with the 722 Research Institute, China Shipbuilding Industry Corporation, Wuhan. He is currently a Postdoctoral Fellow with the Electronic Information School, Wuhan University, Wuhan. His current research interests include hyperspectral imagery, machine learning, and pattern recognition.

JI ZHAO received the B.S. degree in automation from the Nanjing University of Posts and Telecommunication, Nanjing, China, in 2005, and the Ph.D. degree in control science and engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2012. His research interests include computer vision and machine learning.