A Monocular SLAM System Leveraging Structural Regularity in Manhattan World*

Haoang Li[1], Jian Yao[1], Jean-Charles Bazin[2], Xiaohu Lu[1] and Yazhou Xing[1]

***Abstract*— The structural features in Manhattan world encode useful geometric information of parallelism, orthogonality and/or coplanarity in the scene. By fully exploiting these structural features, we propose a monocular SLAM system which can obtain accurate estimation of camera poses and 3D map. The foremost contribution of the proposed system is a structural features based *optimization thread* which contains three novel optimization strategies. First, a rotation optimization strategy using the parallelism and orthogonality of 3D lines is presented. Based on these two geometric cues, we propose a global binding method and an approach for calculating relative rotation to get accurate absolute rotations. Second, a translation optimization strategy leveraging coplanarity is proposed. Coplanar features are effectively identified, and they are exploited by a unified model handling points and lines equivalently to calculate relative translation, followed by obtaining optimal absolute translations. Third, a 3D line optimization strategy utilizing parallelism, orthogonality and coplanarity simultaneously is proposed to obtain an accurate 3D map consisting of structural line segments with low computational complexity. Experiments in man-made environments have demonstrated that the proposed system outperforms existing state-of-the-art monocular SLAM systems in terms of accuracy and robustness.**

I. INTRODUCTION

The goal of simultaneous localization and mapping (SLAM) is to estimate the state of a robot and construct a 3D map of the environment. It is a crucial component for robotics navigation and has been widely studied over the years [1]. Various methodologies for SLAM have been developed based on different sensors including single camera, stereo camera, RGB-D camera, laser scanner and inertial measurement unit (IMU). Among them, monocular SLAM using a single camera has gained considerable popularity in the past decade, due to its inherent advantages: lower price, compactness and simplicity to calibrate [2]. One dominant strategy for SLAM is the feature-based approach. It first detects a set of image features such as points and lines, then matches them between consecutive frames, followed by iteratively estimating the orientation and position of a moving object, as well as reconstructing a 3D map from feature correspondences.

Existing features based methods can be classified into two main categories with respect to the type of features used: *non-structural features* [3]–[7] and *structural features* [8]–[12]. The non-structural features based methods have more universality and robustness, because they can operate in different kinds of environments (e.g. with/without structural regularity), but the accuracy is unsatisfactory without enforcing effective constraints. The structural features based methods consider structural cues, and are often applied in man-made environments which can be abstracted as a set of blocks sharing three common dominant directions in general. In this type of scene, known as Manhattan world, structural regularity can provide effective geometric constraints which are useful to improve the accuracy of SLAM.

On the one hand, non-structural features (without constraints of direction and/or position) based method is relatively mature, and there are numerous monocular SLAM systems which rely on non-structural points [3], non-structural lines [4], or combination of these two types of features [6]. Among them, points based approach is the most popular one, and some representative frameworks [2], [3] perform well in many environments. However, while points can be effectively detected and matched, points based systems are prone to fail in low-textured scenarios like man-made environments where points are insufficient. In addition, as low-level features, points encode less geometric information like parallelism and orthogonality, and can not provide effective constraints, leading to high accumulating errors over time.

To overcome the limitations of points, lines have attracted extensive attention. When there are insufficient points detected in the scene, lines can serve as ideal complements. Representative system using non-structural lines [4] has proved the advantages of lines in environments with less textures. However, a significant drawback of lines is that they are often incompletely detected and partially occluded, leading to the instability of the systems. Previous work on line features [4], [5] seldom considered the structural information of lines, like parallelism, orthogonality and coplanarity, so they are more unstable and yield worse results than points based counterparts under the effect of noise in practical use.

Some researchers have combined non-structural points and lines to make the systems more applicable in some challenging scenes. Pumarola *et al.* [6] proposed a system which can simultaneously process points and lines. This system has demonstrated stability in low-textured environments. Li *et al.* [7] designed a model which handles points and lines efficiently, and can be applied in extreme scenarios with scarce features. However, though these methods can leverage more observations, they fail to overcome the inherent drawbacks of non-structural features introduced above, and their accuracy is limited due to the lack of effective constraints.

On the other hand, structural features based methods

*This work was partially supported by the National Natural Science Foundation of China (Project No. 41571436), and the Hubei Province Science and Technology Support Program, China (Project No. 2015BAA027).

[1]H. Li, J. Yao, X. Lu and Y. Xing are with Computer Vision and Remote Sensing (CVRS) Lab, Wuhan University, P.R. China.

[2]J.C. Bazin is with Computational Media Lab, KAIST, South Korea.

have gained wide attention recently [8]–[12]. In Manhattan world, structural features are mainly reflected in two aspects. Firstly, each 3D structural line is parallel to one of the three mutually orthogonal dominant directions. Under the projective transformation, a set of parallel 3D lines form a cluster of projective lines, which converge at the same point in the image, which is called vanishing point (VP). Secondly, some 3D points and lines are on the same plane whose normal is parallel to one of the three dominant directions.

The first type of structural features – VPs which reflect the parallelism and orthogonality, has been exploited in robotics state estimation. Lee *et al.* [8] proposed a method which uses VPs as virtual landmarks in indoor buildings. Because VPs are observable in the whole scene, the loop closure optimization, i.e., re-visiting a known place to reduce error, can be achieved in non-loop scene. Camposeco and Pollefeys [9] detected VPs using an inertial-aided method, and integrated VPs into a visual odometry system to reduce the angular drift. Inspired by previous work, Zhou *et al.* [10] proposed a system using structural lines of buildings. Their system, which is based on extended Kalman filter (EKF) framework, can alleviate the accumulating orientation errors and outperformed existing work. An important limitation of above systems is that the robustness will be affected when auxiliary information from other sensors is not available, such as laser scanner [8], IMU [9], and wheel odometer [10]. More importantly, the translation error has not been handled effectively. In addition, while error in rotation can be reduced to some extent, there is still room for improvement.

As second type of structural features, coplanar points and lines have relation with both rotation and translation, unlike VPs which can only enforce constraint on rotation. A tracking and mapping method based on coplanar points was proposed in [11], the authors presented an error minimization approach based on coplanarity. Kwon and Lee [12] proposed a particle filtering based SLAM framework using locally planar landmarks, which could gain satisfactory result. However, these methods only consider coplanar points without lines, and the prior knowledge of fixed normals of planes in Manhattan world is also neglected, leading to less stability in indoor low-textured environment like corridors. Other related work on coplanarity have exploited the geometric priors in Manhattan world. Kosecka and Zhang [13] proposed a strategy to extract dominant planes via lines which belong to the same VP, and then recover the camera pose.

Overall, non-structural features can guarantee the robustness, while if structural features are exploited, the accuracy can be improved. Therefore, in this paper, we propose a monocular SLAM system which leverages non-structural and structural features simultaneously. We first exploit non-structural features to obtain rough estimation of camera poses and 3D map following existing methods, and then use the structural features to develop an *optimization thread* containing three novel optimization strategies as main contributions:

- Accurate rotation optimization strategy leveraging the parallelism and orthogonality: A global binding method and an approach for calculating precise relative rotation are proposed to significantly reduce accumulating error of absolute rotations;
- Accurate translation optimization strategy exploiting coplanarity: coplanar features are identified effectively, and then used by a unified model handling coplanar points and lines equivalently to obtain the relative translations, followed by the absolute translations optimization;
- Accurate and efficient 3D map optimization strategy based on parallelism, orthogonality and coplanarity: a novel 3D line parameterization method is designed, along with a reliable cost function based on re-projection error minimization of lines.

Experiments in man-made scenes have shown that the proposed system outperforms existing state-of-the-art monocular SLAM systems in terms of accuracy and robustness.

II. PROBLEM FORMULATION

Throughout this paper, all the matrices, vectors and scalars are denoted as bold capital like "**R**", bold lowercase like "**t**", and plain letters like "i", respectively. Vectors are column-wise in default. We use "$\propto$" to represent equality regardless of scale, and define "$[\cdot]_\times$" as a 3×3 skew symmetric matrix to rewrite cross products by matrix multiplications , i.e.,

$$\left[[x_1, x_2, x_3]^\top\right]_\times = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}.$$

A. Overview of the Proposed System

In the Manhattan world, the proposed system first exploits non-structural features following [6] (other existing approaches based on non-structural features can also be used instead) to obtain *rough* camera poses and a 3D map which inevitably contain the accumulating error over time. To improve the accuracy of the proposed system, we then leverage the structural regularity to develop an optimization thread which contains three novel optimization strategies for rotation, translation and 3D map respectively. Specifically, given an image sequence, rough pose of the i-th camera P_i can be determined based on non-structural features, including absolute rotation matrix $\mathbf{R}_i \in SO3$ and translation vector $\mathbf{t}_i \in \mathbb{R}^3$ which align the camera coordinate $\mathcal{C}_i$ to the world coordinate $\mathcal{W}$, and a rough 3D map in $\mathcal{W}$ is also constructed. To refine $\mathbf{R}_i$, $\mathbf{t}_i$ and the 3D map, the structural features are effectively exploited, which will enforce the following structural constraints: (1) *parallelism* and *orthogonality*; (2) *coplanarity* for optimization.

B. Structural Constraint of Parallelism and Orthogonality

We denote $\mathcal{D} = \{\mathbf{d}_k\}_{k=1}^3$ in $\mathcal{W}$ as a set of three mutually orthogonal dominant directions in Manhattan world. Given a cluster of parallel 3D lines in this scene, they must be aligned with one direction $\mathbf{d}_k$. The parallelism of 3D lines can be reflected by VPs (cf. Section I), i.e, $\mathcal{D}$ correspond to a set of VPs denoted as $\mathcal{V}_i = \{\mathbf{v}_k^i\}_{k=1}^3$ on the image I_i of camera P_i. With the known intrinsic matrix $\mathbf{K}$, the relation between $\mathbf{d}_k$ and $\mathbf{v}_k^i = [vx_k^i, vy_k^i, 1]^\top$ is defined in [14] as

$$\mathbf{v}_k^i \propto \mathbf{K}\mathbf{R}_i\mathbf{d}_k. \tag{1}$$

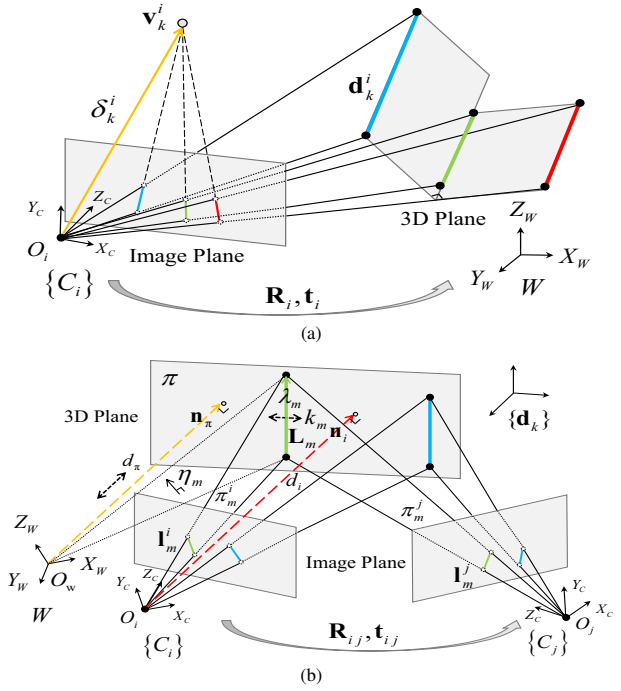


Fig. 1. Illustrations of geometric model of structural features in Manhattan world: (a) structural constraint of parallelism (VD $\boldsymbol{\delta}_k^i$ in yellow is parallel to blue, green and red 3D lines which are all aligned with one dominant direction of world $\mathbf{d}_k^i$); (b) structural constraint of coplanarity (3D lines from $\mathcal{L}$ in green and blue are coplanar).

As shown in Fig. 1(a), we define the vanishing direction (VD) in $\mathcal{C}_i$ as vector $\boldsymbol{\delta}_k^i$ from camera center O_i to $\mathbf{v}_k^i$. The coordinates of VDs can be calculated as $\boldsymbol{\delta}_k^i = \mathbf{K}^{-1}\mathbf{v}_k^i$. VD satisfies an important constraint that $\boldsymbol{\delta}_k^i$ is parallel to $\mathbf{d}_k^i$ which represents $\mathbf{d}_k$ in $\mathcal{C}_i$ [14], i.e.,

$$\boldsymbol{\delta}_k^i \propto \mathbf{d}_k^i \propto \mathbf{R}_i \mathbf{d}_k \tag{2}$$

In addition, the third VD can be computed by cross-product of the other two VDs (since the VDs are orthogonal to each other in Manhattan word).

C. Structural Constraint of Coplanarity

As shown in Fig. 1(b), M 3D lines $\mathcal{L} = \{\mathbf{L}_m\}_{m=1}^M$ expressed by Plücker matrix [14] lie on the same plane $\boldsymbol{\pi}$, satisfying

$$\mathbf{L}_m \boldsymbol{\pi} = 0. \tag{3}$$

We define "coplanar features matches" to reflect coplanarity. The projections of $\mathcal{L}$ to camera P_i and P_j can form line matches $\mathcal{M} = \{\mathbf{l}_m^i, \mathbf{l}_m^j\}_{m=1}^M$. We refer to $\mathcal{M}$ as "coplanar line matches", which share the same homography $\mathbf{H}_{ij}$. Supposing $\mathbf{p}_m^i$ is one arbitrary 2D point on the image line $\mathbf{l}_m^i$, $\mathbf{H}_{ij}$ maps it to its correspondence $\mathbf{p}_m^j$, satisfying $\mathbf{p}_m^j = \mathbf{H}_{ij}\mathbf{p}_m^i$. Because $\mathbf{l}_m^i$ and $\mathbf{l}_m^j$ are correspondences, $\mathbf{p}_m^j$ must lie on $\mathbf{l}_m^j$ as ${\mathbf{l}_m^j}^\top \mathbf{p}_m^j = 0$. Hence, we can obtain

$${\mathbf{l}_m^j}^\top \mathbf{H}_{ij} \mathbf{p}_m^i = 0. \tag{4}$$

Similarly, for image point matches $\mathcal{P} = \{\mathbf{q}_n^i, \mathbf{q}_n^j\}_{n=1}^N$ formed by coplanar 3D points on $\boldsymbol{\pi}$, we refer to $\mathcal{P}$ as "coplanar point matches", satisfying

$$\mathbf{q}_n^j \propto \mathbf{H}_{ij} \mathbf{q}_n^i \tag{5}$$

In the following, we will propose three novel optimization strategies, which are based on the above structural constraints (parallelism and orthogonality are for Section III and V; coplanarity is for Section IV and V).

III. ROTATION OPTIMIZATION

We optimize rotation based on the structural constraint of parallelism and orthogonality. We exploit both global binding and numerous relative rotations $\{\mathbf{R}_{ij}\}$ between camera pairs $\{P_i, P_j\}$ to optimize absolute rotations.

A. VP Extraction and Dominant Directions Initialization

The proposed rotation optimization strategy demands accurate VP information. To obtain VPs, we adopt [15] which can guarantee the globally optimal VPs extraction results and inherently enforce the orthogonality of VDs in Manhattan world. Their main idea is to formulate the VPs extraction task as a consensus set maximization problem, and they solve it by a branch-and-bound procedure (readers are invited to refer to that paper for more details).

Three dominant directions in Manhattan world are fixed as follows. Without lack of generality, the world coordinate system $\mathcal{W}$ is aligned with coordinate system of the first camera $\mathcal{C}_1$. Thus, dominant directions $\mathcal{D}$ can be initialized using VPs $\mathcal{V}_1$ based on (1), i.e., $\mathbf{d}_k \propto \mathbf{I}^{-1}\mathbf{K}^{-1}\mathbf{v}_k^1$, where $\mathbf{I}$ is a 3×3 identity matrix. Note that if less than two VPs are extracted, we discard this frame and test the next one until a qualified frame is found for initialization.

B. Absolute Rotation Optimized by Global Binding

We aim at exploiting "global binding" based on VPs to reduce the accumulating error in $\mathbf{R}_i$ which is originally obtained by non-structural features based method. We define the following cost function related to the constraint (2), and minimize it in Lie algebra:

$$E(\boldsymbol{\omega}_i) = \sum_{k=1}^{3} E_k(\boldsymbol{\omega}_i) = \sum_{k=1}^{3} \arccos\left(\boldsymbol{\delta}_k^i \cdot \mathbf{R}_i \mathbf{d}_k\right), \tag{6}$$

where $\boldsymbol{\delta}_k^i$ and $\mathbf{d}_k$ are unit-norm vectors, symbol "$\cdot$" represents the dot product, and $\boldsymbol{\omega}_i$ in Lie algebra is the mapping from $\mathbf{R}_i$ in Lie group.

The gradient descend method Levenberg-Marquardt (LM) is applied to find the minimum of $E(\boldsymbol{\omega}_i)$, which needs the Jacobian matrix $\mathbf{J}_k = \partial E_k(\boldsymbol{\omega}_i)/\partial \boldsymbol{\omega}_i$. We provide brief derivation of $\mathbf{J}_k$ as follows:

$$\mathbf{J}_k = -\frac{1}{\sqrt{1-\psi^2}} \boldsymbol{\delta}_k^i \frac{\partial \mathbf{d}_k^i}{\partial \boldsymbol{\omega}_i} = \frac{1}{\sqrt{1-\psi^2}} \boldsymbol{\delta}_k^i ([\mathbf{d}_k^i]_\times), \tag{7}$$

where $\psi = \boldsymbol{\delta}_k^i \cdot \mathbf{d}_k^i$.

We treat rough $\mathbf{R}_i$ as initial value for LM method, and obtain the optimized $\hat{\mathbf{R}}_i$ by the time iteration converges. It is worth emphasizing that $\hat{\mathbf{R}}_i$ is obtained based on VPs which can be observed in the global scene, i.e., we bind $\mathcal{C}_i$ to $\mathcal{W}$ directly, so the optimization of P_i is independent from other frames, and the accumulating error in absolute rotation can be effectively reduced.

C. Absolute Rotation Optimized by Relative Rotation

Besides optimizing absolute $\mathbf{R}_i$ using global binding directly, we also leverage relative rotation $\mathbf{R}_{ij}$ between two frames to further refine $\hat{\mathbf{R}}_i$. Specifically, a pair of cameras P_i and P_j is associated by $\mathbf{R}_{ij}$ as $\boldsymbol{\delta}_k^j \propto \mathbf{R}_{ij}\boldsymbol{\delta}_k^i$, and the scale ambiguity can be eliminated using cross product:

$$[\boldsymbol{\delta}_k^j]_\times \mathbf{R}_{ij}\boldsymbol{\delta}_k^i = \mathbf{0}. \tag{8}$$

We parameterize $\mathbf{R}_{ij}$ by quaternion, and rewrite (8) as quadratic equations. Because the rank of $[\boldsymbol{\delta}_k^i]_\times$ is 2, two independent equations can be obtained. Therefore, at least two corresponding VDs can construct a polynomial system containing four quadratic equations with four unknowns (the four variables of the quaternion). We solve this polynomial system using Gröbner basis, which can be easily computed by an automatic generator proposed by Kukelova *et al.* [16]. Note that we can compute numerous $\mathbf{R}_{ij}$ between any pair of two cameras because our method based on VPs does not need the overlap of two images. The significance of $\{\mathbf{R}_{ij}\}$ for optimization of $\{\hat{\mathbf{R}}_i\}$ is introduced as follows.

The number of elements in $\{\mathbf{R}_{ij}\}$ is significantly larger than $\{\hat{\mathbf{R}}_i\}$. To fully exploit the redundancy of $\{\mathbf{R}_{ij}\}$, we adopt the rotation averaging framework [17] which can refine $\{\hat{\mathbf{R}}_i\}$ by $\{\mathbf{R}_{ij}\}$. Specifically, we map $\mathbf{R}_{ij} = \hat{\mathbf{R}}_j\hat{\mathbf{R}}_i^{-1}$ from the Lie group to the Lie algebra by first-order approximation:

$$\boldsymbol{\omega}_{ij} = \boldsymbol{\omega}_j - \boldsymbol{\omega}_i = [\cdots - \mathbf{I} \cdots \mathbf{I} \cdots]\boldsymbol{\omega} = \mathbf{E}\boldsymbol{\omega}, \tag{9}$$

where $\boldsymbol{\omega}_{ij}$ is the mapping from known $\mathbf{R}_{ij}$, $\boldsymbol{\omega}$ contains all unknown $\{\boldsymbol{\omega}_i\}$ to be solved (they are initialized by $\hat{\mathbf{R}}_i$) and $\mathbf{E}$ consists of two identity matrices $\mathbf{I}$ and numerous null matrices. By combining all the observations $\{\boldsymbol{\omega}_{ij}\}$, we can construct an over-determined sparse linear system based on (9). We then solve this system by $L1$ optimizer [18] to compute optimal $\{\boldsymbol{\omega}_i\}$ and finally remap to $\{\mathbf{R}_i\}$.

IV. TRANSLATION OPTIMIZATION

After optimizing the absolute rotations $\{\mathbf{R}_i\}$, in this section we present a novel strategy to refine the absolute translations $\{\mathbf{t}_i\}$ by leveraging an additional structural constraint: the coplanarity of features. We compute normalized relative translation $\tilde{\mathbf{t}}_{ij}$ between cameras P_i and P_j based on coplanar point and/or line matches. Then we retrieve the scale of $\tilde{\mathbf{t}}_{ij}$ for obtaining the optimized absolute $\mathbf{t}_i$.

A. Identification of Coplanar Feature Matches

To optimize the translation based on coplanarity constraint, we first need to determine the coplanar line matches $\{\mathcal{M}_u\}$ and coplanar point matches $\{\mathcal{P}_u\}$ (defined in Section II). Traditional DLT-RANSAC [14] which finds coplanar features without structural constraints is prone to be unstable in the noisy scene. In contrast, our method aims at obtaining $\{\mathcal{M}_u\}$ and $\{\mathcal{P}_u\}$ in a stable and accurate way, even in noisy scene.

In Section III, we have determined line segments clusters $\{\mathcal{S}_k^i\}_{k=1}^3$ of image I_i with respect to their associated VPs. We first match lines between the images I_i and I_j to obtain line cluster correspondences. Without loss of generality, assuming that the line $\mathbf{l}_d^i$ belonging to the line cluster $\mathcal{S}_k^i$ of image I_i, and the line $\mathbf{l}_d^j$ belonging to the line cluster $\mathcal{S}_k^j$ of image I_j are matched, then the line clusters $\mathcal{S}_k^i$ and $\mathcal{S}_k^j$ are associated. Note that for all the line matches from the cluster correspondence $\{\mathcal{S}_k^i, \mathcal{S}_k^j\}$, their corresponding 3D lines are *parallel* but not necessarily *coplanar*, and we consider them as "candidates of $\{\mathcal{M}_u\}$" to be refined. Next, we get the final coplanar line matches $\{\mathcal{M}_u\}$ using "characteristic line" (CL) [19] which is an invariant representation of *coplanar* and *parallel* 3D lines. Specifically, among a set of parallel 3D lines, if some lie on the same plane, their corresponding 2D line matches share a common CL, while other matches related to another 3D plane share a different CL, i.e., each $\mathcal{M}_u$ related to a plane has its own CL. Using different CLs, we can cluster "candidates of $\{\mathcal{M}_u\}$" to obtain final $\{\mathcal{M}_u\}$. Besides, some sets from $\{\mathcal{M}_u\}$ are merged if they share common homography. Finally, point matches are clustered into $\{\mathcal{P}_u\}$ based on existing homographies from $\{\mathcal{M}_u\}$.

B. Normalized Relative Translation from Unified Model

Now we have the coplanar line matches $\{\mathcal{M}_u\}$ and the coplanar point matches $\{\mathcal{P}_u\}$. In the following, we propose a unified model handling coplanar points and lines equivalently to calculate normalized relative translation.

Homography $\mathbf{H}_{i,j}$ of the plane $\boldsymbol{\pi}$ can be decomposed in the following form [14]:

$$\mathbf{H}_{ij} = \mathbf{K}\left(\mathbf{R}_{ij} + \frac{\mathbf{t}_{ij}}{d_i}\mathbf{n}_i^\top\right)\mathbf{K}^{-1}, \tag{10}$$

where d_i is the perpendicular distance from the camera center O_i to the plane $\boldsymbol{\pi}$, and $\mathbf{n}_i$ is the unit normal of the plane $\boldsymbol{\pi}$ in $\mathcal{C}_i$. For brevity, we mark the *normalized* translation $\mathbf{t}_{ij}/d_i$ as $\tilde{\mathbf{t}}_{ij}$. Based on optimized $\mathbf{R}_i$ in Section III, accurate $\mathbf{R}_{ij}$ is determined by $\mathbf{R}_{ij} = \mathbf{R}_j\mathbf{R}_i^{-1}$, and $\mathbf{n}_i$ can be fixed as corresponding $\mathbf{d}_k^i$ calculated by (2).

For a coplanar line match $\{\mathbf{l}_m^i, \mathbf{l}_m^j\} \in \mathcal{M}_u$, we substitute (10) into (4), and rewrite the result as following form:

$$(\mathbf{n}_i^\top\mathbf{K}^{-1}\mathbf{p}_m^i\mathbf{l}_m^{j\top}\mathbf{K})\tilde{\mathbf{t}}_{ij} = \mathbf{l}_m^{j\top}\mathbf{K}\mathbf{R}_{ij}\mathbf{K}^{-1}\mathbf{p}_m^i, \tag{11}$$

which is linear in $\tilde{\mathbf{t}}_{ij}$. An arbitrary point $\mathbf{p}_m^i$ on the line $\mathbf{l}_m^i$ from $\{\mathbf{l}_m^i, \mathbf{l}_m^j\}$ can provide one such equation.

Similarly, for a coplanar point match $\{\mathbf{q}_n^i, \mathbf{q}_n^j\} \in \mathcal{P}_u$, after combining (5) and (10), we can obtain:

$$[\mathbf{q}_n^j]_\times\mathbf{K}(\mathbf{R}_{ij} + \tilde{\mathbf{t}}_{ij}\mathbf{n}_i^\top)\mathbf{K}^{-1}\mathbf{q}_n^i = \mathbf{0}, \tag{12}$$

which can be rewritten as two linear equations regarding $\tilde{\mathbf{t}}_{ij}$, because the rank of $[\mathbf{q}_v^j]_\times$ is two.

Since $\mathcal{M}$ and $\mathcal{P}$ can construct linear equations with respect to $\tilde{\mathbf{t}}_{ij}$ in the same form from (11) and (12), we can combine two types of observations into a unified model. An over-determined linear system can be constructed in the form: $\mathbf{A}\tilde{\mathbf{t}}_{ij} = \mathbf{b}$, and $\tilde{\mathbf{t}}_{ij}$ can be solved by least square method.

C. Absolute Translation Optimized by Relative Translation

We now present how to optimize the absolute translations $\{\mathbf{t}_i\}$ using local normalized relative translations $\{\tilde{\mathbf{t}}_{ij}\}$ which are obtained by exploiting constraint of coplanarity.

We adopt the translation averaging framework [20] which is originally designed for unit-norm relative translation vector, and modify it for our normalized translation $\tilde{\mathbf{t}}_{ij}$ (note that norm of $\tilde{\mathbf{t}}_{ij}$ is not always 1). In the first place, we obtain the scale factor λ_{ij} of each $\tilde{\mathbf{t}}_{ij}$, which is d_i in (10). Note that another significance of retrieving d_i will be introduced in Section V. Once λ_{ij} is obtained, $\mathbf{t}_{i,j}$ can be calculated as $\mathbf{t}_{i,j} = \lambda_{ij}\tilde{\mathbf{t}}_{i,j}$, then ultimate absolute translation $\mathbf{t}_i$ can be determined as follows. For camera pairs $\{P_i, P_j\}$, the relation between the absolute translation $\mathbf{t}_i$ and the relative translation $\mathbf{t}_{ij}$ can be described as

$$-\mathbf{R}_i^\top \mathbf{t}_i + \mathbf{R}_j^\top \mathbf{t}_j = \lambda_{ij}\mathbf{R}_j^\top \tilde{\mathbf{t}}_{i,j}. \tag{13}$$

Collecting numerous observations of $\{\mathbf{t}_{ij}\}$, we can construct an over-determined linear system as $\mathbf{Ht} = \mathbf{g}$, where $\mathbf{H}$ is a sparse matrix, and $\mathbf{t}$ is a set of $\mathbf{t}_i$. This system can be solved using $L1$ optimizer [18] again, and optimal $\{\mathbf{t}_i\}$ are obtained.

V. 3D STRUCTURAL MAP OPTIMIZATION

Using the optimal $\{\mathbf{R}_i\}$ and $\{\mathbf{t}_i\}$ (obtained in Section III and IV), we now present how an accurate 3D structural map can be obtained by considering structural constraint of parallelism, orthogonality and coplanarity simultaneously.

A. Representation of 3D Structural Line

Lines encode more structural information than points, so we express the 3D environment using a "structural map" which consists of 3D structural lines. We use Plücker matrix [14] to represent a 3D line $\mathbf{L}_m$ as

$$\mathbf{L}_m = \begin{bmatrix} [\boldsymbol{\eta}_m]_\times & \boldsymbol{\lambda}_m \\ -\boldsymbol{\lambda}_m^\top & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4},$$

where $\boldsymbol{\eta}_m$ is the normal of the plane formed by $\mathbf{L}_m$ and O_w which is the origin of W, and $\boldsymbol{\lambda}_m$ is the unit direction vector (seen in Fig. 1(b)). Corresponding Plücker coordinate is denoted as $\mathbf{u}_m = [\boldsymbol{\eta}_m^\top, \boldsymbol{\lambda}_m^\top]^\top$. Besides, a 3D plane is expressed as $\boldsymbol{\pi} = [\mathbf{n}_\pi^\top, d_\pi]^\top$, where $\mathbf{n}_\pi$ is the unit normal of $\boldsymbol{\pi}$, and d_π is the perpendicular distance from O_w to $\boldsymbol{\pi}$.

By exploiting the structural constraint including parallelism and coplanarity in Manhattan world, structural lines and dominant plane can be expressed more concisely and precisely. First, based on parallelism constraint, we fix $\boldsymbol{\lambda}_m$ of $\mathbf{L}_m$ and $\mathbf{n}_\pi$ of $\boldsymbol{\pi}$ as corresponding $\mathbf{d}_k$ respectively. Second, we rewrite coplanarity constraint (3) as a linear system with respect to $\boldsymbol{\eta}_m$. Because the rank of $\mathbf{L}_m$ is only 2, which is less than 3 degrees of freedom of $\boldsymbol{\eta}_m$, there is an unknown coefficient k_m in the general solution of $\boldsymbol{\eta}_m$ which also contains undetermined d_π. Therefore, each 3D line $\mathbf{L}_m$ on the plane $\boldsymbol{\pi}$ is expressed as a function regarding respective k_m and common d_π. A parallel geometric illustration of the above algebraic derivation can be expressed as two dashed lines with double-head arrow in Fig. 1(b). Specifically, for the plane $\boldsymbol{\pi}$, we fix its normal $\mathbf{n}_\pi$, so it can only freely move along $\mathbf{n}_\pi$, which is reflected in undetermined d_π. Since $\mathbf{L}_m$ is confined within the plane $\boldsymbol{\pi}$ and its $\boldsymbol{\lambda}_m$ is fixed, it can only slide along the vertical direction of $\boldsymbol{\lambda}_m$ on the plane, which is related to undetermined k_m.

B. 3D Structural Map Optimization

To optimize the 3D line $\mathbf{L}_m$, we need to determine the parameters k_m and d_π of $\mathbf{L}_m$. Our method for 3D structural lines lying on a single plane is presented at first, and it can be easily extended to the case of multiple planes. First, we map $\mathbf{L}_m$ expressed by d_π and k_m to $\mathbf{u}_m$, and transform its coordinate from $\mathcal{W}$ to $\mathcal{C}_i$ using line motion matrix $\mathbf{M}_i$ [21] which consists of known $\mathbf{R}_i$ and $\mathbf{t}_i$, i.e., $\mathbf{u}_m^i = \mathbf{M}_i\mathbf{u}_m$. Then we project $\mathbf{u}_m^i = [\boldsymbol{\eta}_m^{i\top}, \boldsymbol{\lambda}_m^{i\top}]^\top$ onto the image set $\{I_i\}_i^S$ which contains S images participating in the 3D reconstruction of $\mathbf{L}_m$. It is worth noting that the projective line $\hat{\mathbf{l}}_m^i$ on image I_i is only determined by the normal $\boldsymbol{\eta}_m^i$ rather than direction $\boldsymbol{\lambda}_m^i$, i.e., $\hat{\mathbf{l}}_m^i = \bar{\mathbf{K}}\boldsymbol{\eta}_m^i$, where $\boldsymbol{\eta}_m^i$ corresponds to the undetermined parameters k_m and d_π, and $\bar{\mathbf{K}}$ is the known intrinsic matrix for lines [5], which is slightly different from ordinary $\mathbf{K}$. We define the following cost function by minimize re-projection error of lines:

$$\arg\min_{\{k_m\},d_\pi} \sum_{m=1}^{M}\sum_{i=1}^{S}\left(d(\mathbf{s}_m^i, \hat{\mathbf{l}}_m^i) + d(\mathbf{e}_m^i, \hat{\mathbf{l}}_m^i)\right), \tag{14}$$

where $\mathbf{s}_m^i$ and $\mathbf{e}_m^i$ are endpoints of detected image line segment $\mathbf{l}_m^i$, and $d(\cdot,\cdot)$ represents the distance from the detected endpoint to the projective line.

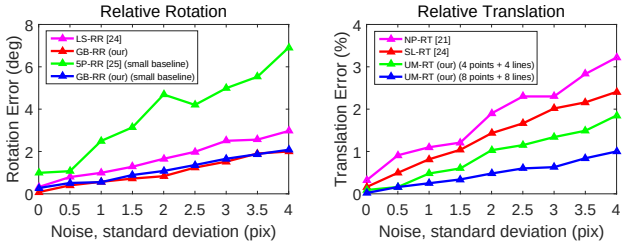
As shown in Fig. 1 (b), we initialize d_π using simple geometric relation between $\mathbf{R}_i$, $\mathbf{t}_i$, and d_i which is determined in Section IV-C. $\boldsymbol{\eta}_m$ of $\mathbf{L}_m$ is initialized by the intersection of $\boldsymbol{\pi}_m^i$ and $\boldsymbol{\pi}_m^j$ which are calculated by $\mathbf{R}_i$, $\mathbf{t}_i$, and $\{\mathbf{l}_m^i, \mathbf{l}_m^i\}$. After minimizing (14), we can determine the optimal parameters k_m and d_π. The 3D endpoints of $\mathbf{L}_m$ are calculated following [5]. For the case of multiple planes, because we have identified coplanar 3D lines in Section IV, we can optimize each coplanar group by (14) independently. Therefore, optimized 3D structural map consisting of line segments from multiple planes can be obtained.

VI. EXPERIMENTS

To demonstrate the performance of the proposed structural features based SLAM system, we conduct experiments on both simulated data and real image sequence. We compare our methods with existing state-of-the-art approaches in terms of accuracy and efficiency. Additional experimental results and source code are available at `http://cvrs.whu.edu.cn/projects/Struct-PL-SLAM/`.

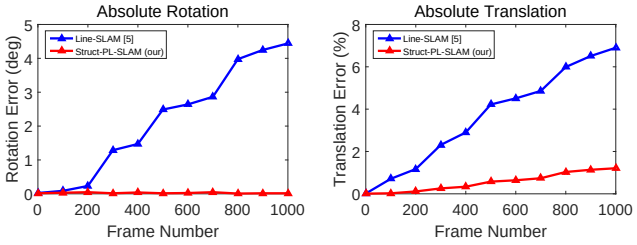
A. Simulated Data

Experiments on simulated data are divided into two parts. First, two images having overlap are synthesized, to compare proposed algorithms with existing methods on relative rotation and translation estimation as well as 3D line segment optimization. Second, a long image sequence is synthesized to make comparisons in large-scale scene. We quantitatively evaluate the accuracy of camera pose using the measure defined in [7]. Specifically, to assess the estimated rotation matrix $\mathbf{R}$, we apply the "rotation error" $E_\mathrm{rot}\,(\mathrm{deg}) = \max_{k=1}^{3}\{\mathrm{acos}(\mathrm{dot}(\mathbf{r}_\mathrm{true}^k, \mathbf{r}^k)) \times 180/\pi\}$, where $\mathbf{r}_\mathrm{true}^k$ and $\mathbf{r}^k$ are the k-th columns of $\mathbf{R}_\mathrm{true}$ and $\mathbf{R}$,



(a) (b)

Fig. 2. Relative pose estimation results between two simulated images with respect to noise: (a) relative rotation; (b) relative translation.



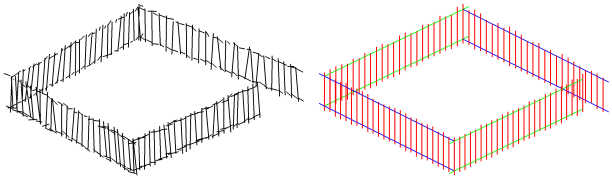
(a) (b)

Fig. 3. Absolute pose estimation results on the long synthetic image sequence: (a) absolute rotation; (b) absolute translation.

TABLE I

A COMPARISON BETWEEN TWO 3D OPTIMIZATION STRATEGIES.

m	**NS-LO** [6]		**S-LO** (our)	
	RMSE	T_{iter}(s)	RMSE	T_{iter}(s)
50	0.251	0.237	0.109	0.032
100	0.227	0.495	0.083	0.077
300	0.274	1.434	0.129	0.213
800	0.286	3.761	0.138	0.585



(a) (b)

Fig. 4. Reconstructed 3D maps consisting of line segments: (a) results of **Line-SLAM** [4]; (b) results of our **Struct-PL-SLAM** (color red, green and blue represent 3 dominant directions).

respectively. The "translation error" to evaluate estimated translation $\mathbf{t}$ is defined as $E_{\text{trans}}\,(\%) = \|\mathbf{t}_{\text{true}} - \mathbf{t}\|/\|\mathbf{t}\| \times 100$. We assess the precision of reconstructed 3D line segment by Hausdorff distance between densely sampled points along the line segment and the ground truth, and the root mean squared error (RMSE) is calculated, following [22].

In a virtual Manhattan world, the simulated 3D points and endpoints of line segments used in the following tests are all distributed into the coordinate interval Ω of $[-2,2] \times [-2,2] \times [4,8]$. By projecting 3D features to cameras, 2D feature matches are constructed. We set the focal length of the virtual camera to 800 pixels, and the principal point is located at the center of image plane which is 640×480 pixels. Cameras are randomly generated on the sphere Φ whose center is at the centroid of Ω and radius is 6. Note that in the following experiments, we test various algorithms under the effect of noise. Specifically, for relative pose estimation experiments, 2D points or endpoints of line segments in feature matches are contaminated by Gaussian noise with varying standard deviations σ from 0.5 to 4 pixels, while in 3D lines optimization test, σ is fixed as 2 pixels. For the experiments with long synthetic image sequence, we set σ to 3 pixels in each frame. The following results we report are based on 1000 independent trials.

a) Relative rotation estimation: For the algorithms to obtain relative rotation based on VPs, we compare our Gröbner basis based method **GB-RR** (cf. Section III-C) with a linear system based method [23] denoted as **LS-RR**. In the interval Ω, we generate 6 3D structural line segments (each two features are parallel to one dominant directions). As shown in Fig. 2(a), **LS-RR** is more likely to be affected by noise than our **GB-RR**, because **LS-RR** is an indirect strategy, i.e., the relative pose is recovered from two rotations whose error is accumulated to the final result. Besides, **LS-RR** parameterizes the rotation by 9 parameters which is redundant, and more unknown variables lead to less robustness. In contrast, our **GB-RR** is a direct method and uses only 4 unknown variables. In addition, the reliable Gröbner basis is exploited for accurate solution.

We also evaluate our **GB-RR** using the same 6 structural lines as above in a nearly degenerate case in which the camera baseline is fixed as a very small value of 0.05. We compare it with 5-point algorithm [24] denoted as **5P-RR** which is provided with 6 point matches (an over-determined case solved by least-squares method). The results reported in Fig.2(a) shows that **5P-RR** is sensitive to the small camera baseline, while our **GB-RR** is more robust. The reason is that **5P-RR** estimates rotation and translation together (extremely small translation would affect rotation's precision), but our **GB-RR** is dedicated to rotation.

b) Relative translation estimation: We compare our method **UM-RT** (cf. Section IV-B) based on the unified model handling coplanar points and lines, with non-structural points based method [20] noted as **NP-RT**, and structural lines based approach [23] denoted as **SL-RT**. Above three methods are provided with the same ground truth relative rotation beforehand. We generate 8 3D points for **NP-RT** and 8 3D structural line segments for **SL-RT**, which are all on the same dominant plane, and we let our **UM-RT** exploit all these 16 features. For a fair comparison on total number of features that are used, the performance of **UM-RT** using 4 3D points and 4 3D lines is also evaluated. As shown in Fig. 2(b), **NP-RT** based on epipolar geometry is highly affected by noise because no structural constraint is enforced. **SL-RT** which exploits the intersection of projective lines has the same idea as **NP-RT** in essence, but structural lines is more robust than points, so more accurate results are obtained. our **UM-RT** using 8 features is more precise than above methods, thanks to the coplanarity constraint. The accuracy of our **UM-RT** leveraging 16 features is highest, because more observations are used to compensate for noise.

c) 3D line optimization: We compare our 3D line optimization strategy **S-LO** based on structural constraint (cf.

Section V) with traditional non-structural constraint based approach [6] denoted as **NS-LO**. Both two methods aim to minimize the re-projection error, and we solve them by the Levenberg-Marquardt method available on the Ceres Solver [25]. We simulate m 3D structural line segments lying on s dominant plane as ground truth. Corresponding image line matches between 2 cameras are corrupted with noise, and then are used to reconstruct m low-accuracy 3D line segments which are treated as object to optimize. Table I reports the optimized results with respect to different number m of 3D lines (s is fixed as 3). We also compare the efficiency by measuring the time cost T_{iter} until the convergence of iteration. It shows that our **S-LO** is more efficient than **NS-LO**, especially when m is large. The reason is that there are only $m + s$ parameters for our **S-LO** to optimize, but **NS-LO** has to consider $6 \times m$ parameters. In terms of accuracy, our **S-LO** has lower RMSE and outperforms **NS-LO** because the structural constraints of direction and position of 3D lines are enforced .

d) Test on long image sequence: To evaluate the proposed system **Struct-PL-SLAM** (cf. Section II-A) based on structural points and lines in the large-scale scene, an experiment is conducted on the long synthetic image sequence. A simulated 3D four-side fence consisting of 292 line segments (25 vertical and 2×24 horizontal ones on each side) and points of the same number is constructed. Around the fence, 800 cameras are generated along a circle which is the intersection of sphere Φ and a horizontal plane passing through the centroid of interval Ω. We synthesize an image sequence by projecting 3D features to cameras. We compare our **Struct-PL-SLAM** with non-structural lines based system [4] denoted as **Line-SLAM**. As shown in Fig. 3, the error accumulation in absolute pose of **Line-SLAM** is significant over time. In contrast, the error of our **Struct-PL-SLAM** remains much lower, demonstrating the advantags of structural features. Fig. 4 shows a comparison of the reconstructed 3D line segments. **Line-SLAM** reconstructs a disordered map, while our **Struct-PL-SLAM** can generate a more accurate and structured map, thanks to precise camera pose and 3D map optimization strategy leveraging structural constraints.

B. Real Images

We evaluate the proposed system on the HRBB4 dataset [26] which is ideal to evaluate visual SLAM systems in the typical corridor scene. The image sequence is recorded by a monocular camera mounted on a moving robot and contains 12,000 frames of 640×320 pixels. The total length of the squared trajectory is about 70 m, and the ground truth of camera positions is provided. As shown in Fig. 5, the proposed system **Struct-PL-SLAM** (cf. Section II-A) can effectively detect and exploit the structural features of the corridor scene. Fig. 5(a) presents the results of VP extraction, and Fig. 5(b) shows the representative coplanar line matches judgment results which are obtained based on the clustering results of 2D line segments with respect to VPs. Note that if a 3D line segment is not parallel to any of the three dominant directions, it will not participate in the camera pose and 3D map optimization. We compare our **Struct-PL-SLAM** with existing state-of-the-art systems:

(a) (b)

Fig. 5. Structural features in the corridor scenario: (a) three orthogonal VPs detection result; (b) four coplanar line matches which corresponding to different 3D planes are determined by one horizontal VP, and marked with respective colors (only one image of the matched image pair is shown).

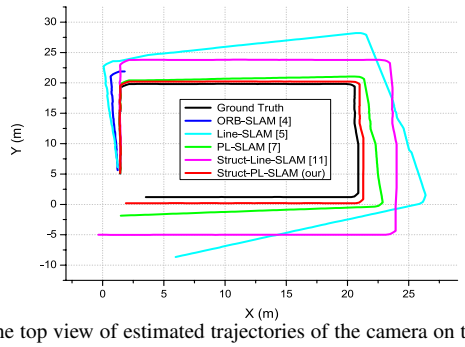


Fig. 6. The top view of estimated trajectories of the camera on the HRBB4 dataset. The black line represents the ground truth. The results of non-structural features based systems: **ORB-SLAM** [3], **Line-SLAM** [4] and **PL-SLAM** [6], as well as structural features based systems **Struct-Line-SLAM** [10] and our **Struct-PL-SLAM** are presented.

- non-structural points based **ORB-SLAM** [3];
- non-structural lines based **Line-SLAM** [4];
- non-structural points and lines based **PL-SLAM** [6];
- structural lines based **Struct-Line-SLAM** [10].

All systems only leverage visual information, so a wheel odometer which is originally exploited by **Struct-Line-SLAM** is not used. In the following, we compare the camera positions and 3D map obtained by various systems.

Fig. 6 shows the trajectories of cameras estimated by various systems. **ORB-SLAM** fails to track at the first corner of the trajectory where point features are extremely scarce, so its trajectory is incomplete. **Line-SLAM** can cover the whole distance, but its error accumulates significantly. Non-structural lines are easily affected by noise due to the lack of effective constraints, and become unstable for practical use. **PL-SLAM** incorporates more feature observations to compensate for noise, and its optimization method is effective, so the system performs better than **ORB-SLAM** and **Line-SLAM**, but the error in rotation is still high. Overall, above non-structural features based systems have unsatisfactory performance. As to structural features based systems, **Struct-Line-SLAM** does not perform as we have expected. Though it can alleviate the accumulating error of rotation to some extent, there is no explicit constraint on translation. Without the aid of wheel odometer for prediction, the scale drift is significant. On the contrary, proposed **Struct-PL-SLAM** has high accuracy and robustness. By enforcing constraint on rotation, its angular error at each bend is smaller than others. For camera position, the result of **Struct-PL-SLAM** has the lowest drift, thanks to the proposed translation optimization strategy leveraging structural constraints.

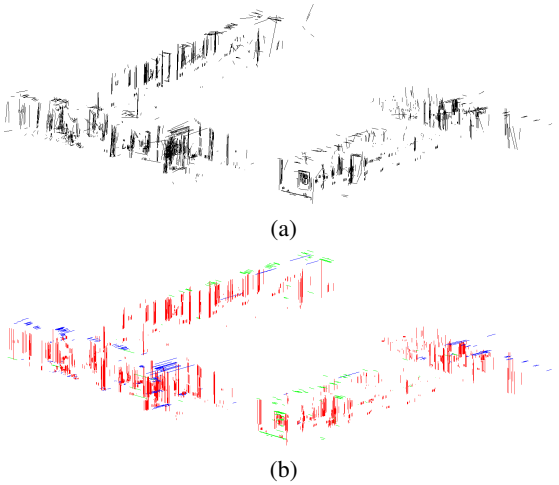


Fig. 7. Comparison between two 3D maps consisting of line segments: (a) result of **PL-SLAM** [6]; (b) result of our **Struct-PL-SLAM** (three dominant directions are marked as color red, green and blue).

Next, we evaluate the accuracy of 3D maps reconstructed by various systems. **ORB-SLAM** fails to generate complete map due to the termination of tracking thread at the first corner of the trajectory. The result of EKF based system **Line-SLAM** and **Struct-Line-SLAM** are relatively unsatisfactory, i.e., lots of 3D line segments deviate far from correct positions. On the contrary, **PL-SLAM** and **Struct-PL-SLAM** can generate superior results. Fig. 7 shows the comparison between 3D map consisting of line segments of **PL-SLAM** and 3D structural map of **Struct-PL-SLAM**. The map of **PL-SLAM** is more disordered, due to limited accuracy of rotation and translation, as well as the noise in image line matches. In contrast, the result of our **Struct-PL-SLAM** is more accurate because it fully exploits the prior knowledge of structural scene for 3D map optimization, and camera pose which are optimized beforehand also contribute to the final result. Therefore, optimized 3D line segments have regular spatial distribution, i.e., they are strictly parallel/orthogonal to each other, and confined in their corresponding 3D plane.

VII. CONCLUSIONS

We proposed a monocular SLAM system based on structural regularity in Manhattan world to obtain accurate camera poses and 3D map. We fully leverage the structural constraints in the following three aspects: (1) parallelism and orthogonality reflected by VPs are exploited to optimize rotations; (2) coplanarity of points and lines is utilized to optimize translations; (3) parallelism, orthogonality and coplanarity of are considered to refine a 3D map. Experiments have shown that our approach outperforms existing state-of-the-art algorithms in terms of accuracy and robustness.

REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tards, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[4] P. Smith, I. Reid, and A. Davison, "Real-time monocular SLAM with straight lines," in *British Machine Vision Conference*, 2006.

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

[6] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "PL-SLAM: Real-time monocular visual SLAM with points and lines," in *IEEE International Conference on Robotics and Automation*, 2017.

[7] H. Li, J. Yao, X. Lu, and J. Wu, "Combining points and lines for camera pose estimation and optimization in monocular visual odometry," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017.

[8] Y. H. Lee, C. Nam, K. Y. Lee, Y. S. Li, S. Y. Yeon, and N. L. Doh, "VPass: Algorithmic compass using vanishing points in indoor environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[9] F. Camposeco and M. Pollefeys, "Using vanishing points to improve visual-inertial odometry," in *IEEE International Conference on Robotics and Automation*, 2015.

[10] H. Zhou, D. Zou, L. Pei, R. Ying, P. Liu, and W. Yu, "StructSLAM: Visual SLAM with building structure lines," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1364–1375, 2015.

[11] C. Mei, S. Benhimane, E. Malis, and P. Rives, "Efficient homography-based tracking and 3-D reconstruction for single-viewpoint sensors," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1352–1364, 2008.

[12] J. Kwon and K. M. Lee, "Monocular SLAM with locally planar landmarks via geometric Rao-Blackwellized particle filtering on Lie groups," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[13] J. Košecká and W. Zhang, "Extraction, matching, and pose recovery based on dominant rectangular structures," *Computer Vision and Image Understanding*, vol. 100, no. 3, pp. 274–293, 2005.

[14] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, second edition, 2003.

[15] J. C. Bazin, Y. Seo, C. Demonceaux, P. Vasseur, K. Ikeuchi, I. Kweon, and M. Pollefeys, "Globally optimal line clustering and vanishing point estimation in Manhattan world," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[16] Z. Kukelova, M. Bujnak, and T. Pajdla, "Automatic generator of minimal problem solvers," in *European Conference on Computer Vision*, 2008.

[17] A. Chatterjee and V. M. Govindu, "Efficient and robust large-scale rotation averaging," in *IEEE International Conference on Computer Vision*, 2013.

[18] E. Candes and J. Romberg, "l1-magic: Recovery of sparse signals via convex programming," 2005, `http://statweb.stanford.edu/~candes/l1magic/downloads/l1magic.pdf`.

[19] C. Kim and R. Manduchi, "Planar structures from line correspondences in a manhattan world," in *Asian Conference on Computer Vision*, 2015.

[20] H. Cui, S. Shen, and Z. Hu, "Robust global translation averaging with feature tracks," in *International Conference on Pattern Recognition*, 2016.

[21] A. Bartoli and P. Sturm, "The 3D line motion matrix and alignment of line reconstructions," *International Journal of Computer Vision*, vol. 57, no. 3, pp. 159–178, 2004.

[22] A. Jain, C. Kurz, T. Thormhlen, and H. P. Seidel, "Exploiting global connectivity constraints for reconstruction of 3D line segments from images," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[23] A. Elqursh and A. Elgammal, "Line-based relative pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[24] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[25] S. Agarwal, K. Mierle, and Others, "Ceres solver," `http://ceres-solver.org`.

[26] Y. Lu, D. Song, and J. Yi, "High level landmark-based visual navigation using unsupervised geometric constraints in local bundle adjustment," in *IEEE International Conference on Robotics and Automation*, 2014.