A Monocular SLAM System Leveraging Structural Regularity in Manhattan World*

Haoang Li1, Jian Yao1, Jean-Charles Bazin2, Xiaohu Lu1, Yazhou Xing1 and Kang Liu1

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 novel monocular SLAM system which provides accurate estimation of camera poses and 3D map. The foremost contribution of the proposed system is a structural feature-based optimization module which contains three novel optimization strategies. First, a rotation optimization strategy using the parallelism and orthogonality of 3D lines is presented. We propose a global binding method to compute an accurate estimation of the absolute rotation of the camera. Then we propose an approach for calculating the relative rotation to further refine the absolute rotation. Second, a translation optimization strategy leveraging coplanarity is proposed. Coplanar features are effectively identified, and we leverage them by a unified model handling both points and lines to calculate the relative translation, and then the optimal absolute translation. 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

Simultaneous localization and mapping (SLAM) aims 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. 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 [1]. 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 the moving camera, as well as reconstructing a 3D map from the feature correspondences.

Existing feature-based methods can be classified into two main categories with respect to the type of features used: non-structural features [2]–[6] and structural features [7]–[11]. The non-structural feature-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 feature-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 feature-based method (i.e. without constraints of direction and/or position) is relatively mature, and numerous monocular SLAM systems rely on non-structural points [2], non-structural lines [3], or combination of these two types of features [5]. Among them, the point-based approach is the most popular one, and some representative frameworks [1], [2] perform well in many environments. However, while points can be effectively detected and matched, the point-based approach is prone to fail in low-textured scenarios like man-made environments where points are insufficient. In addition, as low-level features, points hardly encode geometric information like parallelism and orthogonality, and cannot provide effective constraints, leading to large accumulation error 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. Smith et al. [3] proposed a system using non-structural lines, which has proved the advantages of lines in low-textured environments. 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 [3], [4] seldom considered the structural information of lines, like parallelism, orthogonality and coplanarity, so they are more unstable and yield worse results than point-based methods under the effect of noise.

Some researchers have combined non-structural points and lines to make the systems more applicable in some challenging scenes. Pumarola et al. [5] proposed a system which can simultaneously process points and lines. This system has demonstrated stability in low-textured environments. Li et al. [6] designed a model which handles points and lines efficiently, and can be applied in extreme scenarios with scarce features. However, while 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 feature-based methods have gained wide attention recently [7]–[11]. 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, that is called vanishing point (VP) [12]. Secondly, some 3D points and lines are on the same plane whose normal is parallel to one of the three dominant directions.

As the first type of structural features, VPs reflect the parallelism and orthogonality, and have been exploited in robotics state estimation. Lee et al. [7] 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 [8] detected VPs using an inertial-aided method, and integrated VPs into a visual odometry system to reduce the angular drift. Zhou et al. [9] 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 the above systems is that the robustness is affected when auxiliary information from other sensors is not available, such as laser scanner [7], IMU [8], and wheel odometer [9]. More importantly, the translation error has not been optimized. 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 by Mei et al. [10]. They presented an error minimization approach based on coplanarity. Kwon and Lee [11] proposed a particle filtering-based SLAM framework using locally planar landmarks. 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 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 and structural features have the potential to improve the robustness and accuracy. Therefore, in this paper, we propose a monocular SLAM system leveraging non-structural and structural features simultaneously. We first use non-structural features to obtain a rough estimation of camera poses and 3D map following existing methods, and then exploit the structural features to develop an optimization module composed of three novel optimization strategies. These strategies constitute the main contributions of our paper:

- An 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;
- An accurate translation optimization strategy exploiting coplanarity: coplanar features are identified first, then used by a unified model handling coplanar points and lines equivalently to calculate the relative translation, and then the optimal absolute translation;
- An 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 “r”, and plain letters like “i”, respectively. Vectors are column-wise in default. We use “×” to represent equality regardless of scale, and define “[·]×” as a 3 × 3 skew symmetric matrix to rewrite cross products by matrix multiplications, i.e.,

\[
\begin{bmatrix}
0 & -z_3 & z_2 \\
z_3 & 0 & -z_1 \\
-z_2 & z_1 & 0
\end{bmatrix}_{\times}.
\]

A. Overview of the Proposed System

The proposed system first exploits non-structural features following PL-SLAM [5] (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 error accumulated over time. To improve the accuracy of the proposed system, we then leverage the structural regularity to develop an optimization module which contains three novel optimization strategies for rotation, translation and 3D map respectively. Specifically, given an image sequence, the rough pose of the i-th camera \(P_i\) can be determined based on non-structural features. The camera pose includes the absolute rotation \(R_i \in SO3\) and translation \(t_i \in \mathbb{R}^3\) which align the camera coordinate \(C_i\) to the world coordinate \(W\), and a rough 3D map in \(W\) is also constructed. To refine \(R_i\), \(t_i\) and the 3D map, we leverage the structural features, which will enforce the following structural constraints for optimization:

1. parallelism and orthogonality;
2. coplanarity.

B. Structural Constraint of Parallelism and Orthogonality

We use \(D = \{d_k\}_{k=1}^m\) in \(W\) to denote a set of three mutually orthogonal dominant directions in Manhattan world. Given a cluster of parallel 3D lines in this scene, these lines must be aligned with one direction \(d_k\). The parallelism of 3D lines can be reflected by VPs (cf. Section I), i.e, the set of directions \(D\) is associated with a set of VPs denoted as
Similarly, for image point matches \( P = \{ q_n^i, q_n^j \}_{n=1}^N \) formed by coplanar 3D points on the plane \( \pi \), we refer to \( P \) as “coplanar point matches”, satisfying
\[
q_n^i \propto H_{ij} q_n^j
\]

In the following, we will propose three novel optimization strategies, which are based on the above structural constraints (parallelism and orthogonality are for Sections III and V; coplanarity is for Sections 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 \( \{ R_{ij} \} \) between camera pairs \( \{ P_i, P_j \} \) to optimize absolute rotations.

A. VP Extraction and Dominant Directions Initialization

The proposed rotation optimization strategy leverages VP information. To obtain VPs, we adopt the line-based approach [15] which guarantees the globally optimal VPs estimation and inherently enforces the orthogonality of VPs in Manhattan world. Their main idea is to formulate the VP 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 \( W \) is aligned with the coordinate system of the first camera \( C_1 \). Thus, dominant directions \( D \) can be initialized using VPs \( V_1 \) based on (1), i.e., \( d_k \propto I^{-1} K^{-1} v_k \), where \( I \) is the \( 3 \times 3 \) 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 error accumulation of \( R \), whose initial solution obtained by non-structural feature-based method (cf. Section II-A). We define the following cost function related to the constraint (2), and minimize it in Lie algebra:
\[
E(\omega_i) = \sum_{k=1}^{3} E_k(\omega_i) = \sum_{k=1}^{3} \arccos(\delta_k^i \cdot R_i d_k),
\]
where \( \delta_k^i \) and \( d_k \) are unit-norm vectors, symbol “\( \cdot \)” represents the dot product, and \( \omega_i \) in Lie algebra is the mapping from \( R_i \) in Lie group.

The gradient descent method Levenberg-Marquardt (LM) is applied to find the minimum of \( E(\omega_i) \), which needs the Jacobian matrix \( J_k = \partial E_k(\omega_i) / \partial \omega_i \). We provide brief derivation of \( J_k \) as follows:
\[
J_k = -\frac{1}{\sqrt{1 - \psi^2}} \delta_k^i \frac{\partial d_k^i}{\partial \omega_i} = \frac{1}{\sqrt{1 - \psi^2}} \delta_k^i (\langle d_k^i \rangle),
\]
where \( \psi = \delta_k^i \cdot d_k^i \).

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

### C. Absolute Rotation Optimized by Relative Rotation

Besides optimizing the absolute $R_i$ using global binding directly, we also leverage the relative rotation $R_{ij}$ between two frames to further refine $R_i$. Specifically, a pair of cameras $P_i$ and $P_j$ is associated by $R_{ij}$ as $\delta_i^j \propto R_{ij} \delta_i^k$, and the scale ambiguity can be eliminated using cross product:

$$[\delta_i^j] \times R_{ij} \delta_i^j = 0. \quad (8)$$

We parameterize $R_{ij}$ by quaternion, and rewrite (8) as quadratic equations. Under the presence of noise, these equations cannot be strictly satisfied. We combine them to construct a least-square cost function. After taking partial derivatives with respect to the four variables of the quaternion, we obtain a polynomial system containing four equations. We solve this polynomial system using Gröbner basis, which can be easily computed by an automatic generator [16]. Note that we can compute numerous $R_{ij}$ between any pair of two cameras since our method based on VPs does not need the overlap of two images. The significance of $\{R_{ij}\}$ for optimization of $\{R_i\}$ is introduced as follows.

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

$$\omega_{ij} = \omega_j - \omega_i = [\cdots - I \cdots I \cdots] \omega = E \omega, \quad (9)$$

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

### IV. TRANSLATION OPTIMIZATION

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

#### A. Identification of Coplanar Feature Matches

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

In Section III, we have determined the line segments clusters $\{S^1_i\}_{i=1}^3$ of image $I_i$ with respect to their associated VPs. We first match line images between the images $I_i$ and $I_j$ to obtain line cluster correspondences by $V$-junction descriptor [19]. Without loss of generality, assuming that the line $l^i_i$ belonging to the line cluster $S^1_i$ of image $I_i$, and the line $l^j_i$ belonging to the line cluster $S^1_j$ of image $I_j$ are matched, then the line clusters $S^1_i$ and $S^1_j$ are associated. Note that for all the line matches from the cluster correspondence $\{S^1_i,S^1_j\}$, their corresponding 3D lines are parallel but not necessarily coplanar, and we consider them as “candidates of $\{M_u\}$” to be refined. Next, we get the final coplanar line matches $\{M_u\}$ using “characteristic line” (CL) [20] 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 $M_u$ related to a plane has its own CL. Using different CLs, we can cluster the “candidates of $\{M_u\}$” to obtain the final $\{M_u\}$. Besides, some sets from $\{M_u\}$ are merged if they share a common homography. Finally, point matches are clustered into $\{P_u\}$ based on existing homographies from $\{M_u\}$.

#### B. Normalized Relative Translation from Unified Model

Based on the coplanar line matches $\{M_u\}$ and the coplanar point matches $\{P_u\}$, we propose a unified model handling coplanar points and lines equivalently to calculate the normalized relative translation.

The homography $H_{ij}$ of the plane $\pi$ can be decomposed in the following form [14] :

$$H_{ij} = K \left( R_{ij} + \frac{t_{ij}}{d_i} n_i^\top \right) K^{-1}, \quad (10)$$

where $d_i$ is the perpendicular distance from the camera center $O_i$ to the plane $\pi$, and $n_i$ is the unit normal of the plane $\pi$ in the camera frame $C_i$. For brevity, we mark the normalized translation $t_{ij}/d_i$ as $t_{ij}$. Based on the optimized $R_i$ in Section III, the accurate $R_{ij}$ is determined by $R_{ij} = R_i R_j^{-1}$, and $n_i$ can be fixed as corresponding $d_i^j$ calculated by (2).

For a coplanar line match $\{l^j_i, l^j_m\} \in M_u$, we substitute (10) into (4), and rewrite the result in the following form:

$$ (n_i^\top K^{-1} p_j^i m^j m^j K) t_{ij} = l_i^j m^j K R_{ij} K^{-1} p_m^j, \quad (11)$$

which is linear in $t_{ij}$. An arbitrary point $p_m^j$ on the line $l_m^j$ from $\{l_m^j, l_m^i\}$ can provide one such equation.

Similarly, for a coplanar point match $\{q_n^i, q_n^m\} \in P_u$, after combining (5) and (10), we can obtain:

$$ [q_n^j]^\top K (R_{ij} + t_{ij} n_i^\top) K^{-1} q_n^i = 0, \quad (12)$$

which can be rewritten as two linear equations regarding $t_{ij}$ because the rank of $[q_n^j]^\top$ is two.

Since $M$ and $P$ provide linear equations with respect to $t_{ij}$ in the same form from (11) and (12), we can combine points and lines into a unified model. An over-determined
linear system can be constructed in the form: \( \mathbf{A} t_{ij} = \mathbf{b} \), and \( t_{ij} \) can be solved by least square method.

**C. Absolute Translation Optimized by Relative Translation**

We now present how to optimize the absolute translations \( \{t_i\} \) using the local normalized relative translations \( \{t_{ij}\} \) which are obtained by exploiting the coplanarity constraint.

We adopt the translation averaging framework [21] which is originally designed for unit-norm relative translation vector, and modify it for our normalized translation \( t_{ij} \) (note that the norm of \( t_{ij} \) is not always 1). First, we obtain the scale factor \( \lambda_{ij} \) of each \( t_{ij} \), which is \( d_l \) in (10). Note that another significance of retrieving \( d_l \) will be introduced in Section V. Once \( \lambda_{ij} \) is obtained, \( t_{ij} \) can be calculated as \( t_{ij} = \lambda_{ij} t_{ij} \), then the ultimate absolute translation \( t_i \) can be determined as follows. For camera pairs \( \{P_i, P_j\} \), the relation between the absolute translation \( t_i \) and the relative translation \( t_{ij} \) can be described as

\[ -\mathbf{R}_i^\top t_i + \mathbf{R}_j^\top t_j = \lambda_{ij} \mathbf{R}_i^\top \hat{t}_{ij}. \]

Collecting numerous observations of \( \{t_{ij}\} \), we can construct an over-determined linear system as \( \mathbf{H} \mathbf{t} = \mathbf{g} \), where \( \mathbf{H} \) is a sparse matrix, \( \mathbf{t} \) is a set of \( \mathbf{t} \), and \( \mathbf{g} \) consists of \( \lambda_{ij} \mathbf{R}_i^\top \hat{t}_{ij} \). This system can be solved using L1 optimizer [18] again, and the optimal \( \{t_i\} \) are obtained.

**V. 3D STRUCTURAL MAP OPTIMIZATION**

Using the optimal \( \{\mathbf{R}_i\} \) and \( \{t_i\} \) (obtained in Sections 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

\[ \mathbf{L}_m = \begin{bmatrix} \eta_m^\top & \lambda_m \end{bmatrix} \in \mathbb{R}^{4 \times 4}, \]

where \( \eta_m \) is the normal of the plane formed by \( \mathbf{L}_m \) and \( O_w \) which is the origin of \( W \), and \( \lambda_m \) is the unit direction vector (shown in Fig. 1(b)). The corresponding Plücker coordinate is denoted as \( \mathbf{u}_m = \begin{bmatrix} \eta_m^\top & \lambda_m \end{bmatrix}^\top \). Besides, a 3D plane is expressed as \( \mathbf{P}_m = \begin{bmatrix} \eta_m^\top & \lambda_m \end{bmatrix}^\top \), where \( \mathbf{u}_m \) is the unit normal of \( \mathbf{P}_m \), and \( d_m \) is the perpendicular distance from \( O_w \) to \( \mathbf{P}_m \).

By exploiting the structural constraint including parallelism and coplanarity in Manhattan world, structural lines and dominant planes can be expressed more concisely and precisely. First, based on the parallelism constraint, we fix \( \lambda_m \) of \( \mathbf{L}_m \) and \( \mathbf{u}_m \) of \( \pi \) as corresponding \( \mathbf{d}_k \) respectively. Second, we rewrite coplanarity constraint (3) as a linear system with respect to \( \eta_m \). Because the rank of \( \mathbf{L}_m \) is only 2, which is less than 3 degrees of freedom of \( \eta_m \), there is an unknown coefficient \( k_m \) in the general solution of \( \eta_m \) which also contains undetermined \( d_m \). Therefore, each 3D line \( \mathbf{L}_m \) on the plane \( \pi \) is expressed as a function regarding respective \( k_m \) and common \( d_m \). A 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 \( \pi \), we fix its normal \( \mathbf{n}_n \), so it can only freely move along \( \mathbf{n}_n \), which is reflected in undetermined \( d_m \). Since \( \mathbf{L}_m \) is contained within the plane \( \pi \) and its \( \lambda_m \) is fixed, it can only slide along the vertical direction of \( \lambda_m \) 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 \( d_m \) 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 \( d_m \) and \( k_m \) to \( \mathbf{u}_m \), and transform its coordinate from \( W \) to \( C_i \) using line motion matrix \( \mathbf{M}_{i} \) [22] which consists of known \( \mathbf{R}_i \) and \( \mathbf{t}_i \), i.e., \( \mathbf{u}_m = \mathbf{M}_i \mathbf{u}_m \).

Then we project \( \mathbf{u}_m \) onto the image set \( \{I_i\} \) which contains \( S \) images participating in the 3D reconstruction of \( \mathbf{L}_m \). It is worth noting that the projective line \( \hat{\mathbf{I}}_m \) on image \( I_i \) is only determined by the normal \( \eta_m^i \), rather than the direction \( \lambda_m^i \), i.e., \( \hat{\mathbf{I}}_m = \mathbf{K} \eta_m^i \), where \( \eta_m^i \) corresponds to the undetermined parameters \( k_m \) and \( d_m \), and \( \mathbf{K} \) is the known intrinsic matrix for lines [4], which is slightly different from ordinary \( \mathbf{K} \). We define the following cost function by minimizing the re-projection error of lines:

\[ \arg \min_{k_m, d_m} \sum_{m=1}^{M} \sum_{i=1}^{S} \left( d(s_m^i, \hat{\mathbf{I}}_m^i) + d(e_m^i, \hat{\mathbf{I}}_m^i) \right), \]

where \( s_m^i \) and \( e_m^i \) are the two endpoints of the 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 \( d_m \) using simple geometric relation between \( \mathbf{R}_i \), \( \mathbf{t}_i \), and \( d_l \) which is obtained in Section IV-C. \( \eta_m^i \) of \( \mathbf{L}_m \) is initialized by the intersection of \( \pi_m^i \) and \( \pi_m^i \), which are calculated by \( \mathbf{R}_i \), \( \mathbf{t}_i \), and \( \{\mathbf{l}_m^i, \mathbf{l}_m^j\} \). After minimizing (14), we can determine the optimal parameters \( k_m \) and \( d_m \). The 3D endpoints of \( \mathbf{L}_m \) are calculated by segment trimming [4]. 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, we can finally obtain the optimized 3D structural map consisting of line segments from multiple planes.

**VI. EXPERIMENTS**

To demonstrate the performance of the proposed structural feature-based SLAM system, we conduct experiments on both synthesized 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. Synthesized Data**

Experiments on synthesized data are divided into two parts. First, two images with overlap are synthesized, to
We 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 [6]. Specifically, to assess the estimated rotation matrix $\mathbf{R}$, we apply the “rotation error” $E_{\text{rot}}(\deg) = \max_{k=1}^3 \{ \text{acos} (\text{dot}(\mathbf{r}_{k,\text{true}}, \mathbf{r}_{k})) \times 180/\pi \}$, where $\mathbf{r}_{k,\text{true}}$ and $\mathbf{r}_{k}$ are the $k$-th columns of $\mathbf{R}_{\text{true}}$ and $\mathbf{R}$, respectively. The “translation error” to evaluate the 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 the 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) [23] is calculated.

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 $\Omega = [-2, 2] \times [-2, 2] \times [4, 8]$. 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 whose size is $640 \times 480$ pixels. Cameras are randomly generated on the sphere $\Phi$ whose center is at the centroid of $\Omega$ 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 $\sigma$ from 0.5 to 4 pixels, while in 3D lines optimization test, $\sigma$ is fixed as 2 pixels. For the experiments with long synthetic image sequence, we set $\sigma$ 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 [24] denoted as LS-RR. In the interval $\Omega$, we generate 6 3D structural line segments (each two features are parallel to one dominant direction). 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 [25] 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 point-based method [21] denoted as NP-RT, and structural line-based approach [24] denoted as SL-RT. The three methods above 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 the 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 are 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 the 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. Table I

<table>
<thead>
<tr>
<th>$m$</th>
<th>NS-LO [5]</th>
<th>S-LO (our)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE</td>
<td>$T_{\text{iter}}$ (s)</td>
</tr>
<tr>
<td>50</td>
<td>0.251</td>
<td>0.237</td>
</tr>
<tr>
<td>100</td>
<td>0.277</td>
<td>0.495</td>
</tr>
<tr>
<td>300</td>
<td>0.274</td>
<td>1.434</td>
</tr>
<tr>
<td>800</td>
<td>0.286</td>
<td>3.761</td>
</tr>
</tbody>
</table>

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 $\Omega = [-2, 2]\times[-2, 2]\times[4, 8]$. 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 whose size is $640 \times 480$ pixels. Cameras are randomly generated on the sphere $\Phi$ whose center is at the centroid of $\Omega$ 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 $\sigma$ from 0.5 to 4 pixels, while in 3D lines optimization test, $\sigma$ is fixed as 2 pixels. For the experiments with long synthetic image sequence, we set $\sigma$ 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 [24] denoted as LS-RR. In the interval $\Omega$, we generate 6 3D structural line segments (each two features are parallel to one dominant direction). 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 [25] 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 point-based method [21] denoted as NP-RT, and structural line-based approach [24] denoted as SL-RT. The three methods above 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 the 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 are 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 the 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 the non-structural constraint-based approach [5] denoted as NS-LO. Both two methods aim to minimize the re-projection error, and we solve them by the Levenberg-Marquardt method available in the Ceres Solver\(^1\). 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 \(t_{\text{iter}}\) until convergence. 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, we conducted an experiment on a long synthetic image sequence. A simulated 3D four-side fence consisting of 292 line segments (25 vertical and \(2 \times 24\) 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 the sphere \(\Phi\) and a horizontal plane passing through the centroid of interval \(\Omega\). We synthesize an image sequence by projecting 3D features to cameras. We compare our Struct-PL-SLAM with the non-structural line-based system [3] 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 advantages 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 the structural constraints.

B. Real Images

We evaluate the proposed system on the HRBB4 dataset [26]. The image sequence is recorded in a typical corridor scene by a monocular camera mounted on a moving robot and contains 12,000 frames of \(640 \times 320\) pixels. The

\(^1\)http://ceres-solver.org

length of the squared trajectory is about 70 m, and the ground truth camera positions are provided. The end of the sequence does not overlap with the beginning, so there is no test on loop closure. 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. Fig. 5(a) presents the results of VP extraction; Fig. 5(b) shows the representative coplanar line matches results obtained by 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:

- non-structural point-based ORB-SLAM [2];
- non-structural line-based Line-SLAM [3];
- non-structural points and line-based PL-SLAM [5];
- structural line-based Struct-Line-SLAM [9].

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 camera trajectories 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 it performs better than ORB-SLAM and Line-SLAM, but the error in rotation is still high. Overall, the above non-structural feature-based systems have unsatisfactory per-
formance. As to structural feature-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 our Struct-PL-SLAM has the lowest drift, thanks to the proposed translation optimization strategy leveraging structural constraints.

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 the 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. CONCLUSION

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 lines are used 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