DIRECT SPARSE VISUAL ODOMETRY WITH STRUCTURAL REGULARITIES FOR LONG CORRIDOR ENVIRONMENTS

Fanjin Cheng[1], Chun Liu[1,*], Hangbin Wu[1], Mengchi Ai[1,2]

[1] College of Surveying and Geo-informatics, Tongji University, 200092 Shanghai, China -
(cfj, liuchun, hb, aimengchi)@tongji.edu.cn
[2] Information Sciences and Technology, The Pennsylvania State University, State College, USA - mxa1097@psu.edu

KEY WORDS: Visual Odometry, Simultaneous Localization and Mapping, Manhattan World, Structural Lines

ABSTRACT:

Simultaneous Localization and Mapping are the key requirements for many practical applications of robotics. However, traditional visual approaches rely on features extracted from textured surfaces, so they barely work well in indoor scenes (e.g. long corridors containing large proportions of smooth walls). In this work, we propose a novel visual odometry method to overcome these limitations, which integrates structural regularities of man-made environments in a direct sparse visual odometry system. By fully exploiting structural lines that align with the dominant direction in the Manhattan world, our approach becomes more accurate and robust to texture-less indoor environments, specially, long corridors. Given a series of image inputs, we first use the direct sparse method to obtain the coarse relative pose between camera frames, and then calculate vanishing points on each frame. Secondly, we use structural lines as rotation constraints, and perform a sliding window optimization to reduce both photometric and rotation errors, to further improve the trajectory accuracy. Through the benchmark test, it is proved that our method performs better than that of the existing visual odometry approach in long corridor environments.

1. INTRODUCTION

Accurately estimating the position and orientation of an agent in an indoor scene is a challenging problem, that is usually addressed by Simultaneous Localization and Mapping (SLAM) technologies (Bailey, Durrant-Whyte, 2006). SLAM has become a very active research field due to its wide application in autonomous driving, 3D reconstruction, AR and VR. Of all SLAM technologies, visual SLAM has become the most popular in recent years. In contrast to a complete SLAM pipeline, visual odometry (VO), which tracks the camera's pose from a series of images (without global optimization such as loop closure and relocalization), often drifts over time (Fraundorfer, Scaramuzza, 2012). By introducing extra constraints such as IMU measurements, VO has also shown competitive performance against SLAM systems (Qin et al., 2018), and can be performed at a high frame rate. However, visual tracking in unknown environments still presents some challenges. These challenges are when operating in texture-less environments, it is often necessary to add extra constraints due to little visual information, in order to reduce the drift of trajectory estimation.

VO algorithm can be divided into two categories. (i)Feature-based method, in which features are extracted and descriptors are calculated and stored for matching between frames, while most of the image information is discarded (Mur-Artal, Tardós, 2017). (ii)Direct method, in which camera poses are optimized based on photometric and/or geometric errors rather than feature correspondences (Engel et al., 2018). Although feature-based methods perform well in textured environments, they are unstable in those environments with fewer or repeated textures, such as long corridors, due to their over dependence on visual features. Feature extraction and matching steps also introduce more error sources. In contrast, direct methods skip the feature selection step, and the constraint comes from an overall camera pose. Therefore, even if a single point cannot offer enough information, it can also rely on other points to correct the geometric relationship, so as to find the correct projection point, thus obtain enough image information even in a texture-less long corridor environment. There is some VO methods that leveraging line information as extra constraints (Yijia et al., 2018). Although these Point-Line systems showed promising results, the trajectory estimation is still unstable due to the influence of occlusions on line matching.

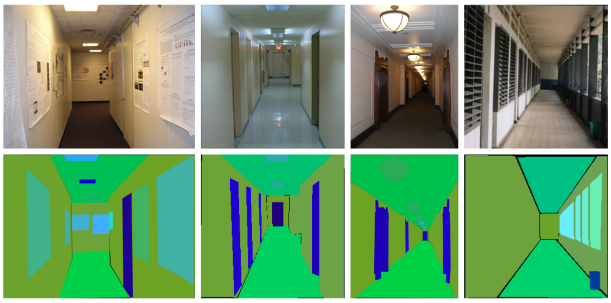

Figure 1. Typical structural scene – the long corridor. The ceiling and the floor are usually texture-less, and the floor is rarely stacked with objects due to its function. Repeated textures of the doors and posters are confusing to visual tracking. Feature-based method cannot find enough and reliable correspondences. Corridors are usually narrow so the structural information can be easily observed in a single image.

Architectural scenes, having planes, texture-less walls, sharp angles and axially aligned geometries, often exhibit strong structural regularity, including parallelism and orthogonality, as shown in Figure 1 (Zhou et al., 2019). The existence of these structures provides an opportunity to constrain and simplify pose estimation. Such scenes can be abstracted as Manhattan world (Coughlan, Yuille, 1999). It states that all the planes

[*] Corresponding author

of the world are aligned in three dominant directions, that is, the world is a piecewise-axis-aligned planar. The advantage of using this structural regularity in a VO system is obvious: parallel lines aligned with the Manhattan world create direction constraints that prevent local direction errors from growing. It has been applied to indoor modeling (Furukawa et al., 2009) and scene understanding (Mulam et al., 2010). With the help of Manhattan world assumptions, the robustness and accuracy of visual SLAM have been improved (Li et al., 2018) (Zhou et al., 2015).

Based on the above ideas, we tried to exploit the structural regularity of man-made scene to the direct VO system. In this work, We described the combination in detail, including Manhattan world representation, structural lines parameterizationand error terms designing. We conducted experiments on the open benchmark dataset. Results show that our method achieves better performance by combining structure regularities of the structured environment. The main contributions of this work are described below:

1. We seamlessly integrates the Manhattan world hypothesis with the most advanced framework of sparse direct method.
2. We designed new error terms to merge the structural information in local sliding window optimizations which are performed on several keyframes to refine the camera pose and the point depth.

2. RELATED WORK

The known limitation of feature-based approach is that the matching step is error-prone, and subsequent filters are often required to remove outliers and features with uneven spatial distribution. In contrast, the direct method does not require feature matching, but directly uses sensor inputs, such as image intensity, to optimize the cost function in order to determine the relative camera motion. The direct method can also be categorized into dense and sparse method. The advantage of the dense method is that it can utilize all available information in the image and generate a dense map useful for robot navigation.The direct dense method has been evolved in (Steinbrücker et al., 2011), RGB-D SLAM (Kerl et al., 2013), LSD-slam (Engel et al., 2014), etc. Since the information contained in the image is highly redundant, the direct sparse method tries to minimize the photometric error only at sparse random points on the image, so as to improve the efficiency and speed (Engel et al., 2018). The advantage of the sparse method is that there are fewer points, so the calculation cost is usually small, which can save a lot of time during multiple iterations.

The structural regularity of artificial environments is characterized by the dominance of line features in such environmens. The line also has the advantage of robustness to the illumination change. Some feature-based systems attempt to optimize the camera pose by using straight lines as a complement to the point feature. For example, PL-SLAM (Pumarola et al., 2017) which integrates line features in visual SLAM and PL-VIO (Yijia et al., 2018) combined with visual-inertial odometry(VIO). However, in the long corridor environment with almost no texture, the corners are scarce and there are a lot of repeated features, so the method based on point and lines often introduces more ambiguities. Studies have shown that adopting line features in SLAM systems can sometimes lead to worse performance than using points only (Zhou et al., 2015).

Another manifestation of the structural regularity in man-made environment is that structure lines are aligned with three dominant directions. Therefore, if the structure line aligned with the dominant direction is found, the direction information can be used to constrain the camera direction, and such direction information is shown as the vanishing point of parallel lines in the image. It has been shown that the use of vanishing point can improve trajectory accuracy. Li et al. (Li et al., 2018) used vanishing points to reduce error accumulation in Monocular SLAM. (Zhou et al., 2015) used both directional information and structural lines as feature constraints. All of the above methods are enhanced on the basis of the feature-based method. According to the characteristics of long corridors, it is a logical direction to use structural constraints to expand the direct method, but as far as we know, there is no direct VO system using structural constraints, and we are the first to establish such a pipeline.

3. SYSTEM OVERVIEW

We use structural regularities in the VO system. The proposed system first use DSO (Engel et al., 2018) to obtain a rough inter-frame camera pose with cumulative errors over time. In order to improve the accuracy of the system, we use the regularity of Manhattan world, and add the direction constraint to the sliding window optimization , and then optimize the result. Figure 2 shows the main components of our System: (i) Visual Odometry, and (ii) Sliding Window Optimization. We receive image data and pass it to the system, extract the line in the visual odometry step and calculate the direction of the vanishing point, then operate initialization procedure and coarse tracking which roughly estimating the relative camera motion between the current frame and last key frame, then update the preset depth value and save the Hessian matrix. After that, we pass the state variable and rotation constraints to the back-end, then judge whether the current frame is a key frame or not according to the key frame selection strategy. If it is a key frame, it will be added into the sliding window optimization module, then old map points from previous N-1 keyframes will be projected to the current keyframe, creating photometric residuals. At the same time, according to the rotation constraint implied by the vanishing point, the relative rotation matrix between previous N-1 keyframes and the current keyframe are concerned for optimizing absolute rotation, and thus refine the camera pose.

In the remaining sections, we will describe the main components in more detail.

4. VISUAL ODOMETRY

4.1 Line Segment Detection and Vanishing Point Estimation

In order to make use of the vanishing point information, for a single image, we first use LSD line detector (Grompone von Gioi et al., 2010) to detect the image lines. With the obtained line set, we use the branch-and-bound framework in conjunction with the rotational search space (Bazin et al., 2012). In this way, the consensus set of the line in the dominant direction is maximized to ensure the best estimation of the vanishing point.

We use vector $\mathbf{u}_i$ and $\mathbf{v}_i$ to represent the i^{th} pair of lines in a line set, and the rotation matrix $\mathbf{R}$ as their relationship. $\angle(\mathbf{u}, \mathbf{v})$ represents the angle between vector $\mathbf{u}$ and $\mathbf{v}$, in $[0, \pi]$. When there

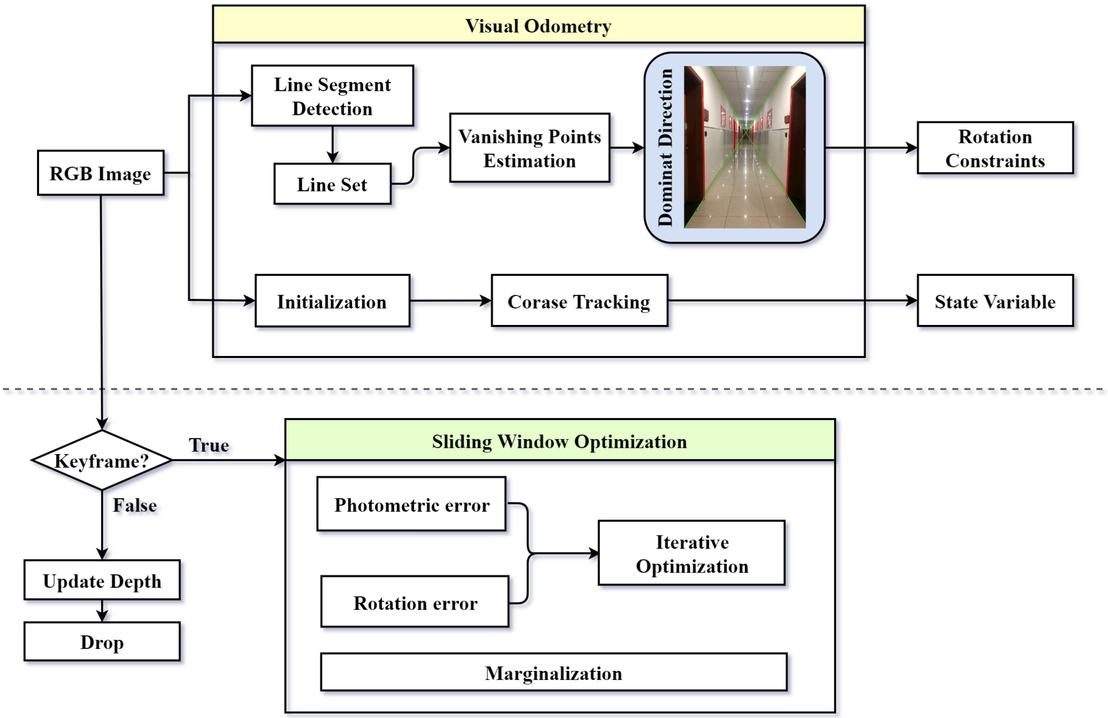

Figure 2. Overview of the proposed VO pipeline.

is zero noise and no outliers, any i^{th} pair satisfies $\mathbf{R}\mathbf{u}_i = \mathbf{v}_i$, and $\angle(\mathbf{R}\mathbf{u}_i, \mathbf{v}_i) = 0$. When there is noise and outliers, we define that the pair $(\mathbf{u}_i, \mathbf{v}_i)$ is an inlier when the angle difference is less than the residual tolerance δ, i.e. $\angle(\mathbf{R}\mathbf{u}_i, \mathbf{v}_i) \leq \delta$.

The problem of maximizing the uniform set of the rotation model is expressed as follows. If the i^{th} pair is an inner point, $y_i = 1$, otherwise $y_i = 0$. By adjusting $\mathbf{R}$, we maximize the sum of y_i and then the number of interior points:

$$\begin{aligned} \max_{y,R} \quad & \sum_{i}^{N} y_i \\ \text{s.t.} \quad & y_i \angle(\mathbf{R}\mathbf{u}_i, \mathbf{v}_i) \leq y_i \delta, \forall i = 1 \dots N \\ & y_i \in \{0, 1\}, \forall i = 1 \dots N \\ & \mathbf{R} \in SO(3) \end{aligned} \tag{1}$$

We use the branch-and-bound (B&B) framework and the rotational space search method to divide the defined interval of the model to be estimated into smaller subspaces, then discard or ine them. Thus, the size of the subspace decreases iteratively, the estimated solution converges to the optimal solution, and stops when the desired precision is reached.

The Manhattan hypotheses are as follows:(i)Each 3D structure line is parallel to each other in a certain dominant direction, and the 2D lines which are projected to the camera plane by 3D parallel lines will generate a vanishing point.(ii)The line between the vanishing point and the camera center is parallel to the 3D parallel lines.

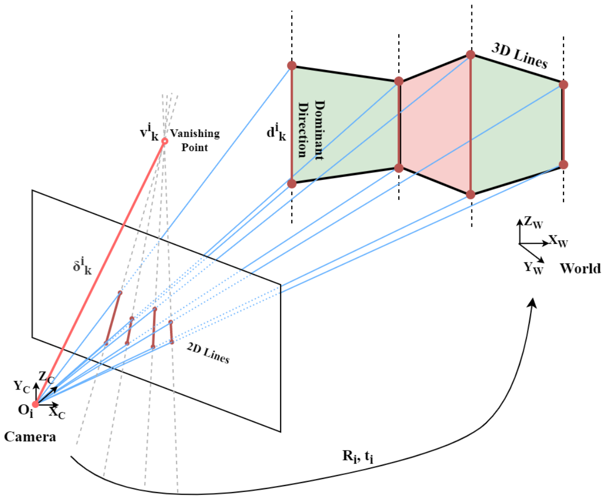

Figure 3. Geometric model of structure lines in Manhattan world.

As shown in Figure 3, we use $\mathbf{d}_k, k = 1 \dots 3$ to represent the three orthogonal dominant directions in the Manhattan world coordinate system. According to hypothesis(i): a set of 3D parallel lines in the scene must be in the same direction with one of the dominant directions, $\mathbf{d}_k$. Since parallel lines intersect at the infinity point, the parallelism between 3D lines can be represented by a vanishing point, thus the relation between vanishing point and dominant direction can be obtained as:

$$\mathbf{v}_k^i \propto \mathbf{K}\mathbf{R}_i\mathbf{d}_k \tag{2}$$

where $\mathbf{v}_k^i = \left[vx_k^i, vy_k^i, 1\right]^T$ is the vanishing point, three vanishing points on image $\mathbf{I}_i$ are $\mathbf{V}_i = \left\{\mathbf{v}_k^i\right\}_{k=1}^3$. $\propto$ represents the equality regardless the scale factor. $\mathbf{K}$ is the camera intrinsic, $\mathbf{R}_i$ is the absolute rotation matrix of frame i.

According to hypothesis(ii): the line between the vanishing point and the camera center is parallel to the 3D parallel lines.

We define δ_k^i as the vector from the camera center to the vanishing point. Its relation with the vanishing point $\mathbf{v}_k^i$ is $\delta_k^i = \mathbf{K}^{-1}\mathbf{v}_k^i$. By combining Eq.(2), the rotation constraint can be obtained as:

$$\delta_k^i \propto \mathbf{d}_k^i \propto \mathbf{R}_i \mathbf{d}_k \tag{3}$$

4.1.1 Error Function and Jacobian Calculation: According to the constraint of Eq.(2), the cost function can be constructed and minimized in the subsequent optimization step:

$$E\left(\omega_i\right) = \sum_{k=1}^{3} E_k\left(\omega_i\right) = \sum_{k=1}^{3} \arccos\left(\delta_k^i \cdot \mathbf{R}_i \mathbf{d}_k\right) \tag{4}$$

where δ_k^i and $\mathbf{d}_k$ are unit vectors and ω_i is the 3D rotation vector corresponding to the rotation matrix $\mathbf{R}_i$. Then the jacobian matrix can be obtained as (Li et al., 2018):

$$\begin{aligned} \mathbf{J}_k &= \partial E_k\left(\omega_i\right) / \partial \omega_i \\ &= -\frac{1}{\sqrt{1-\psi^2}} \delta_k^i \frac{\partial \mathbf{d}_k^i}{\partial \omega_i} = \frac{1}{\sqrt{1-\psi^2}} \delta_k^i \left(\left[\mathbf{d}_k^i\right]_\times\right) \end{aligned} \tag{5}$$

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

4.2 Initialization and Coarse Tracking

For the first camera frame, the system generates the intrinsic $\mathbf{K}$ of the image pyramid and extracts a certain number of points layer by layer satisfying a certain uniform distribution law. The inverse depth of the point is initialized to 1, and the neighborhood relation of adjacent points of each point on the image plane is constructed by KD tree. Then initialization step is performed when the later frames have arrived.

When a map point is observed by the camera frame at the beginning, we only know its 2D image coordinates and the depth is unknown. Such points are called immature points. As the camera moves, the tracking process tracks these immature map points on each image frame to determine the inverse depth of each point and its range of variation. If the inverse depth of the point converges during this process, we consider it as a mature point and store the inverse depth parameter of the point and the Hessian information representing the local gradient. Each mature map point also needs to indicate its host frame, indicating that the point is obtained by back projection of this frame.

Each 3D point, starting from a host frame, is projected to a target frame by multiplying the depth value, thus establishing a projection residual. As long as the residuals are within a reasonable range, we can assume that these points are projected by the same point. All mature map points can be projected on any target frame except the host frame to form a residual term. The stack addition of all the residuals constitutes an optimization problem to be solved at the optimization step. Due to motion and occlusion, not every mature point can be successfully projected to any other frame, so we also need to set the state of each point: valid/marginalized/invalid.

Coarse Tracking: According to (Engel et al., 2018), a number of candidate pose are set as the initial values of relative camera motion from the previous keyframe to the current frame. These initial values are set based on assumptions such as static and constant speed referring to the pose of the previous two frames and the last key frame. Then the system starts tracking using these initial values from the top of the image pyramid, and if it finds a suitable initial prediction, it jumps out of the loop. Then tracking step calculates the best pose from the coarse to the fine. In the coarse tracking step, the photometric error is calculated without changing the pose, and returned the cumulative error. Then the variables needed for subsequent calculation of jacobian matrix are saved.

4.2.1 Error Function and Jacobian Calculation: We define the host frame as $\mathbf{I}_1$, and target frame as $\mathbf{I}_2$. The relative motion from the host coordinate system to the target coordinate system is $\xi_{21} \in se(3)$. Suppose that a pixel point $\mathbf{p}_1$ in the host frame has an inverse depth of ρ_1 (initialized to 1) and camera intrinsic $\mathbf{K}$, the 3D point coordinate $\mathbf{P}_1$ in the host coordinate system is:

$$\mathbf{P}_1 = \pi^{-1}\left(\mathbf{p}_1\right) = \mathbf{K}^{-1}\mathbf{p}_1/\rho_1 \tag{6}$$

where $\pi^{-1}(x)$ is an inverse projection transformation. The coordinate of pixel point $\mathbf{p}_2$ corresponding to $\mathbf{p}_1$ in the target frame can be obtained as follows: $\mathbf{p}_2 = \pi\left(\exp\left(\xi_{21}^{\Lambda}\right)\mathbf{P}_1\right) = \pi\left(\exp\left(\xi_{21}^{\Lambda}\right)\pi^{-1}\left(\mathbf{p}_1\right)\right)$.

Therefore, according to the assumption of photometric invariance, the residual of the corresponding pixel points in the host frame and the target frame is:

$$r\left(\mathbf{p}_1\right) = I_2\left(\mathbf{p}_2\right) - \exp(a) I_1\left(\mathbf{p}_1\right) - b \tag{7}$$

where a, b are the brightness transfer function parameters to increase the robustness of the system to light, $\exp(a)$ is equivalent to $\frac{t_j \exp\left(a_j\right)}{t_i \exp\left(a_i\right)}$, t_i, t_j is the exposure time of two frames respectively. When t_1 and t_2 are given, the initial value of a is $a = \log\left(t_2/t_1\right)$.

In order to increase the degree of point differentiation and facilitate calculation, 8 points around the central point are selected as a patch and these 8 points share the same inverse depth of the central point (Engel et al., 2018). After the weighted sum of the errors of each point, the residual error function of the one patch is obtained (Huber norm form):

$$\begin{aligned} E_{\mathbf{p}} &= \sum_{\mathbf{p}_1 \in \mathcal{N}(\mathbf{p})} w_{\mathbf{p}} \left\| I_2\left(\mathbf{p}_2\right) - \exp(a) I_1\left(\mathbf{p}_1\right) - b \right\|_\gamma \\ &= \sum_{\mathbf{p}_1 \in \mathcal{N}(\mathbf{p})} w_{\mathbf{p}} H\left(r\left(\mathbf{p}_1\right)\right) \end{aligned} \tag{8}$$

where $w_{\mathrm{p}} = \frac{c^2}{c^2 + \|\nabla I(\mathbf{p})\|_2^2}$ is used to reduce the weight of points with high gradients.

For all points in the current frame that can be projected to the target frame, their errors are added up to obtain the total photometric error function:

$$E = \sum E_{\mathrm{p}} \tag{9}$$

After constructing the objective function of the least square problem, in order to use the Gauss-Newton method to solve this problem, we carry out the first-order Taylor expansion of the error function to find the optimal state increment, so that the entire error function gradually declines until convergence. Since the Huber kernel function is used, to satisfy the least square problem squared terms, we use the error function:

$$f(\mathbf{x}) = \sqrt{w_h} r = \sqrt{w_h} \left(I_2 \left(\mathbf{p}_2 \right) - \exp(a) I_1 \left(\mathbf{p}_1 \right) - b \right) \quad (10)$$

where $w_h = \begin{cases} 1 & , |r| < \sigma \\ \sigma / |r| & , |r| >= \sigma \end{cases}$, and σ is a constant.

We set state variable $\mathbf{x} = \left[\rho_1^{(1)}, \ldots, \rho_1^{(N)}, \epsilon^T, a, b \right]^T_{(N+8) \times 1} = [\mathbf{x}_a, \mathbf{x}_\beta]^T$, where $\rho_1^{(1)}, \ldots, \rho_1^{(N)}$ is the inverse depth of N points in the host frame, ϵ is the pose increment between two images (6dof), a and b are the brightness parameters. In order to optimize the error, the jacobian matrix of the error function on the state variable is required to be calculated. The initialization step needs to solve for N+8 parameters, while the subsequent optimization process only needs 8 parameters. The points whose inverse depth Jacobian does not meet certain conditions will be filtered out and subsequent inverse depth updates will not be carried out. We use $\mathbf{J}_\alpha = \left[\frac{\partial f(\mathbf{x})}{\partial \rho_1^{(1)}}, \ldots, \frac{\partial f(\mathbf{x})}{\partial \rho_1^{(N)}} \right] 1 \times N$ to represent the jacobian matrix of the error function with respect to the variables $\mathbf{x}_\alpha$, and $\mathbf{J}_\beta = \left[\frac{\partial f(\mathbf{x})}{\partial \epsilon}, \frac{\partial f(\mathbf{x})}{\partial a}, \frac{\partial f(\mathbf{x})}{\partial b} \right]_{1 \times 8}$ to variables$\mathbf{x}_\beta$. The total Jacobian matrix can be written as $\mathbf{J} = [\mathbf{J}_\alpha, \mathbf{J}_\beta]_{1 \times (N+8)}$, more details can be refer to (Engel et al., 2018).

5. SLIDING WINDOW OPTIMIZATION

In order to improve the optimization efficiency, we use the sliding window optimization method, that the system only optimizes keyframes in the window (we set the window size to 7). Therefore, only the vanishing point of the keyframe can impose rotation constraint on the direction, which speeds up the system.

5.1 Key Frame Selection and Marginalization

When the weighted sum of the changes of optical flow between the current camera frame and the previous key frame, the changes of optical flow without considering the rotation and the changes of exposure parameters is greater than 1, a new key frame is created. Then this frame is added to the sliding window, and the dimensions of Hessian matrix and b are extended. If the number of key frames is greater than the window size, one of the previous key frames will be selected and the frame and the points contained in it will be removed, and the dimension of Hessian matrix and b will be reduced. This process also involves transferring the information of the deleted frames and points to the remaining frames in the window. Such step is called marginalization. We use Schur complement to marginalize the old variables, ensuring the sparse structure of Hessian matrix.

If the current frame is considered to be a non-keyframe, it will be used to update inverse depth of immature points in all previous key frames in the window. If the current frame is considered to be a key frame, it will be used to update inverse depth either and then passed into the sliding window. All mature points in previous key frames and immature points that conforms to certain conditions are used to establish a new error. After that, the error will be added to the total energy function.

5.2 Absolute Rotation Optimization using Relative Rotation

If we represent the absolute 3D rotation of frame i relative to the global coordinate system as $\mathbf{R}_i$, then the relative rotation $\mathbf{R}_{ij}$ between frame i and j can be calculated as:

$$\mathbf{R}_{ij} = \mathbf{R}_j \mathbf{R}_i^{-1} \quad (11)$$

We define a set of global rotations as $\mathbf{R}_{global} = \{\mathbf{R}_1, \cdots, \mathbf{R}_N\}$, according to Eq.(11), The problem becomes fitting the global rotation to minimize the distance between the observed relative rotation and the calculated relative rotation from the global rotation:

$$\arg \min_{\mathbf{R}_{global}} \sum_{(i,j) \in \mathcal{E}} d^2 \left(\mathbf{R}_{ij}, \mathbf{R}_j \mathbf{R}_i^{-1} \right) \quad (12)$$

We use $\omega \in so(3)$ to represent a 3D rotation vector and define all the rotations as $\omega_{global} = [\omega_1, \cdots, \omega_N]^T$, then the first-order approximation of the relative rotation $\mathbf{R}_{ij} = \mathbf{R}_j \mathbf{R}_i^{-1}$ can be written as:

$$\omega_{ij} = \omega_j - \omega_i = \underbrace{[\cdots - \mathbf{I} \cdots \mathbf{I} \cdots]}_{\mathbf{A}_{ij}} \omega_{global} \quad (13)$$

We can conclude all the relations to:

$$\mathbf{A} \boldsymbol{\omega}_{global} = \boldsymbol{\omega}_{rel} \quad (14)$$

where ω_{rel} is a vector stacked by all the relative rotations ω_{ij}, and $\mathbf{A}$ is stacked by all the$\mathbf{A}_{ij}$. Consider the existence of outliers in the relative rotation observed in Lie algebra in any given iteration, we have $\Delta \omega_{rel} = \mathbf{A} \Delta \omega_{global} + e$. Thus we use L1 optimizer instead of L2 to minimize $\|\mathbf{A} \Delta \omega_{global} - \Delta \omega_{rel}\|_{\ell_1}$. After that, we use the L1 results as an initial input to the iteratively reweighted least squares(IRLS) method (Chatterjee, Govindu, 2013), in which each $\mathbf{R}_{ij}$ estimate is appropriately weighted. The objective function can be defined as

$$\begin{gathered} \min_{\mathbf{x}} E = \min_{\mathbf{x}} \sum_i \rho \left(\|\mathbf{e}_i\| \right) = \min_{\mathbf{x}} \sum_i \frac{\mathbf{e}_i^2}{\mathbf{e}_i^2 + \sigma^2} \\ \Rightarrow \frac{\partial E}{\partial \mathbf{x}} = \frac{\partial E}{\partial \mathbf{e}} \frac{\partial \mathbf{e}}{\partial \mathbf{x}} = 0 \\ \Rightarrow \mathbf{A}^T \boldsymbol{\Phi}(\mathbf{e}) \mathbf{A} \mathbf{x} = \mathbf{A}^T \boldsymbol{\Phi}(\mathbf{e}) \mathbf{b} \end{gathered} \quad (15)$$

where $\rho(x) = \frac{x^2}{x^2 + \sigma^2}$ is the cost function, σ is a coordinate parameter, $\Phi(\mathbf{e})$ is a diagonal matrix, $\Phi(i,i) = \frac{\sigma^2}{\left(e_i^2 + \sigma^2 \right)^2}$. In order to obtain the $\mathbf{x}$ that minimize the cost, we take turns to estimate Φ and $\mathbf{x}$ until convergence, which gives a global optimization result of absolute rotation.

5.3 Objective Function Optimization

To optimize the frame and point information in the sliding window, we iterated over all the variables in the sliding window using the Gauss-Newton method. We expand the error function to the first order at current state:

$$f(\mathbf{x} + \Delta\mathbf{x}) \approx f(\mathbf{x}) + \mathbf{J}\Delta\mathbf{x} \tag{16}$$

then replace it in the energy function, and calculate the differentiation of variables:

$$\frac{\partial \frac{1}{2} f^2(\mathbf{x} + \Delta\mathbf{x})}{\partial \Delta\mathbf{x}} = f(\mathbf{x}+\Delta\mathbf{x}) \frac{\partial f(\mathbf{x} + \Delta\mathbf{x})}{\partial \Delta\mathbf{x}} \approx (f(\mathbf{x})+\mathbf{J}\Delta\mathbf{x})\mathbf{J} \tag{17}$$

Incremental equation can be obtained by setting Eq.(17) to 0:

$$\mathbf{J}^T\mathbf{J}\Delta\mathbf{x} = -\mathbf{J}^T f(\mathbf{x}) \tag{18}$$

where $\Delta\mathbf{x}$ is the overall update quantity. We have

$$\mathbf{H}\Delta\mathbf{x} = \mathbf{g} \tag{19}$$

where $\mathbf{H} = \sum \mathbf{J}^T\mathbf{J}$, $\mathbf{g} = -\sum \mathbf{J}^T f(\mathbf{x})$. The optimal increment is obtained by solving the incremental equation, and the obtained increment is used to update the state $\mathbf{x} \leftarrow \mathbf{x} + \Delta\mathbf{x}$. Then we recalculate the error with the new state variable, then compare the new error with the old one, and consider whether to accept the optimization.The previous optimization provides a priori for the next step, and the iterative solution is carried out.

6. EXPERIMENTS

In order to evaluate the effectiveness of our proposed method, we conducted comparative experiments using the TUM Visual-Inertial Dataset (Schubert et al., 2018). The dataset provides camera images captured through long corridor environments with 10241024 resolution at 20 Hz, and IMU measurements at 200Hz. We chose two typical indoor sequences from the dataset, corridor 01 and corridor 04, both of which contain a long corridor. We use OpenCV to realize fast line segment extraction. All the experiments were run on a computer, Intel NUC 6i7KYK, which was set up with Ubuntu 16.04, with a Intel i7-6700HQ CPU and 32-GB RAM.

As the dataset only provides groundtruth at the start and the end of one sequence, we evaluated the performance using mean relative position error (RPE) with Sim(3) alignment. The accuracy of our proposed method was compared with the state-of-the-art direct VO methods: DSO (Engel et al., 2018), to validate the advantages of the proposed method. DSO is a typical visual odometry of direct sparse method, which uses photometric residuals to estimate the depth of points and camera motion between frames. These results are shown in Table 1, which indicate that our method outperforms DSO.

Dataset	DSO(m)	Ours(m)
Corridor 01	0.494038	0.307068
Corridor 04	0.107506	0.072928

Table 1. Comparison of Performance(mean RPE)

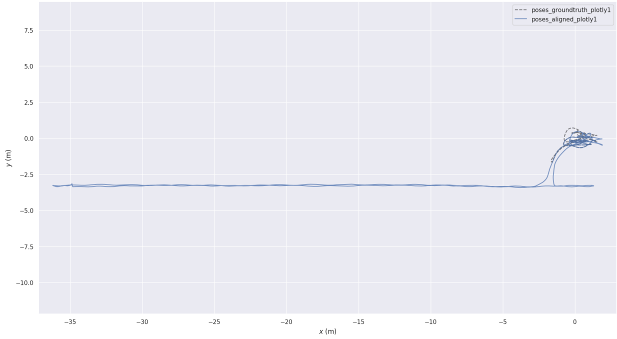

Figure 4. Trajectory of corridor 04.

Figure 4 are trajectory graphs drawn on the proposed method based on sequence corridor 04 using evo evaluation tool (Grupp, 2017). We also aligned and visualized the trajectories of the two methods with the ground truth reference. Figure 5 shows the aligned trajectory of sequence corridor 04. Note that in this diagram, the intermediate straight line segment is caused by the lack of ground truth data in the sequence. It can be seen that, even in the room at the start and end of the track, our method can improve the trajectory using structural constraints .

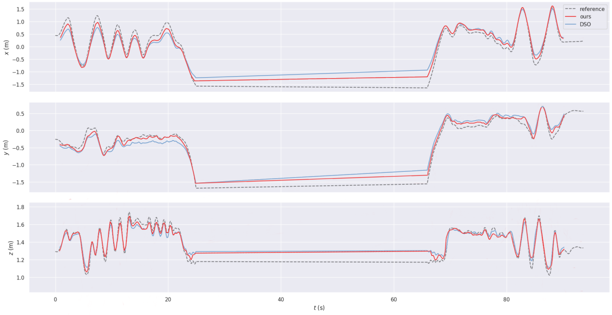

Figure 5. The aligned trajectory of DSO and ours on x, y, z axis separately.

7. CONCLUSION

In this paper, we propose a novel direct sparse visual odometry system based on structural constraints in man-made buildings, aiming at improving the trajectory accuracy of the existing VO system in long corridors. This system uses structural lines to optimize VO rotation. Experimental results on public datasets demonstrate the effectiveness and competitiveness of our approach. Note that our method will perform better when combined with global optimization methods such as loop closure, because our proposed VO system provides a better VO outputs than other algorithms. This increase in accuracy allows SLAM systems to reduce their reliance on computationally expensive global optimizations. In the future work, we will further improve the optimization model and consider the semantic information such as the planar constraints of the building.

REFERENCES

Bailey, T., Durrant-Whyte, H., 2006. Simultaneous localization and mapping (SLAM): part II. *IEEE Robotics Automation Magazine*, 13(3), 108-117.

Bazin, J.-C., Seo, Y., Pollefeys, M., 2012. Globally optimal consensus set maximization through rotation search. 7725, 539–551.

Chatterjee, A., Govindu, V. M., 2013. Efficient and robust large-scale rotation averaging. *2013 IEEE International Conference on Computer Vision*, 521–528.

Coughlan, J. M., Yuille, A. L., 1999. Manhattan world: compass direction from a single image by bayesian inference. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 2, 941–947 vol.2.

Engel, J., Koltun, V., Cremers, D., 2018. Direct Sparse Odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3), 611-625.

Engel, J., Schoeps, T., Cremers, D., 2014. Lsd-slam: large-scale direct monocular slam. 8690, 1–16.

Fraundorfer, F., Scaramuzza, D., 2012. Visual Odometry : Part II: Matching, Robustness, Optimization, and Applications. *IEEE Robotics Automation Magazine*, 19(2), 78-90.

Furukawa, Y., Curless, B., Seitz, S. M., Szeliski, R., 2009. Manhattan-world stereo. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 1422–1429.

Grompone von Gioi, R., Jakubowicz, J., Morel, J., Randall, G., 2010. LSD: A Fast Line Segment Detector with a False Detection Control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4), 722-732.

Grupp, M., 2017. evo: Python package for the evaluation of odometry and slam.

Kerl, C., Sturm, J., Cremers, D., 2013. Dense visual slam for rgb-d cameras. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2100–2106.

Li, H., Yao, J., Bazin, J., Lu, X., Xing, Y., Liu, K., 2018. A monocular slam system leveraging structural regularity in manhattan world. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2518–2525.

Mulam, H., Efros, A., Hebert, M., 2010. Blocks world revisited: Image understanding using qualitative geometry and mechanics. 482–496.

Mur-Artal, R., Tardós, J. D., 2017. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*, 33(5), 1255-1262.

Pumarola, A., Vakhitov, A., Agudo, A., Sanfeliu, A., Moreno-Noguer, F., 2017. Pl-slam: Real-time monocular visual slam with points and lines. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 4503–4508.

Qin, T., Li, P., Shen, S., 2018. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics*, 34(4), 1004-1020.

Steinbrücker, F., Sturm, J., Cremers, D., 2011. Real-time visual odometry from dense rgb-d images. *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 719–722.

Yijia, H., Zhao, J., Guo, Y., He, W., Yuan, K., 2018. PL-VIO: Tightly-Coupled Monocular Visual–Inertial Odometry Using Point and Line Features. *Sensors*, 18, 1159.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A., 2019. Semantic Understanding of Scenes Through the ADE20K Dataset. *International Journal of Computer Vision*, 127, 302–321. doi.org/10.1007/s11263-018-1140-0.

Zhou, H., Zou, D., Pei, L., Ying, R., Liu, P., Yu, W., 2015. StructSLAM: Visual SLAM With Building Structure Lines. *IEEE Transactions on Vehicular Technology*, 64(4), 1364-1375.

ACKNOWLEDGEMENTS

This work was supported by the National Key RD Program of China (2016YFB0502102), National Key RD Program of China (2018YFB1305003), and the International Exchange Program for Graduate Students, Tongji University.