Hindawi Publishing Corporation
Scientific Programming
Volume 2016, Article ID 5369780, 8 pages
http://dx.doi.org/10.1155/2016/5369780

Research Article

A Novel Metric Online Monocular SLAM Approach for Indoor Applications

Yongfei Li, Shicheng Wang, Dongfang Yang, and Dawei Sun

High-Tech Institute of Xi'an, Xi'an, Shaanxi 710025, China

Correspondence should be addressed to Yongfei Li; lyfei314@163.com

Received 9 May 2016; Revised 29 July 2016; Accepted 7 August 2016

Academic Editor: Wenbing Zhao

Monocular SLAM has attracted more attention recently due to its flexibility and being economic. In this paper, a novel metric online direct monocular SLAM approach is proposed, which can obtain the metric reconstruction of the scene. In the proposed approach, a chessboard is utilized to provide initial depth map and scale correction information during the SLAM process. The involved chessboard provides the absolute scale of scene, and it is seen as a bridge between the camera visual coordinate and the world coordinate. The scene is reconstructed as a series of key frames with their poses and correlative semidense depth maps, using a highly accurate pose estimation achieved by direct grid point-based alignment. The estimated pose is coupled with depth map estimation calculated by filtering over a large number of pixelwise small-baseline stereo comparisons. In addition, this paper formulates the scale-drift model among key frames and the calibration chessboard is used to correct the accumulated pose error. At the end of this paper, several indoor experiments are conducted. The results suggest that the proposed approach is able to achieve higher reconstruction accuracy when compared with the traditional LSD-SLAM approach. And the approach can also run in real time on a commonly used computer.

1. Introduction

In the robotics community, simultaneous localization and mapping (SLAM) refers to creating the surrounding map and determining self-position, which is necessary for a robot to autonomously navigate in an unknown environment [1]. The map of the unknown environment must be built incrementally. This means the class of methods must focus on computational algorithms which integrate the new information incrementally [2]. Because camera sensor is cheap, is light, and has low power requirement when compared with other sensors like depth camera, visual SLAM has become a popular research topic.

Traditional visual navigation usually uses a stereo visual system, which can directly provide the 3-dimensional information of circumstance, and the position of cameras can be easily estimated by utilizing the visual difference coming from two or more cameras. Whereas the accuracy of stereo visual navigation is limited by the length of base line, this problem is crucial especially in applications where the base line is seriously limited, such as remote sensing and micro UAVs. Therefore, the monocular visual navigation tends to be more general and commonly used.

Generally speaking, there exist two classes of monocular SLAM: feature-based methods and direct methods. In feature-based approaches, including filtering-based [3, 4] and keyframe-based, the geometric information is estimated from image sequences. The whole process is usually split into two sequential steps: extracting feature observations from the image and calculating the scene geometry and camera pose as a function of these feature observations only by using multiview stereo matches.

This uncoupling predigests the overall problem at the cost of information lose, such as information presenting curved edges. This class of image information often makes up a large part of the image especially in man-made environment and is important for a robot to fulfill tasks like obstacle avoidance.

Direct visual odometry (VO) methods overcome this limitation by calculating camera pose directly on the image intensities, in which all information contained in the image is

used. Moreover, more geometry information of the environment can be used in the direct methods, and that is helpful for obtaining higher accuracy and robustness, especially in simplex environments where few key points are available. The geometry information about the scene is valuable for robotics in many applications such as augmented reality. It is well known that direct image alignment is well established for stereo sensors or RGB-D [5, 6]. However, in monocular visual applications, the existed scale ambiguity problem is hard to solve by direct approach. Until recently [7–9], the precise and intact dense depth maps are computed with a variation formulation. However, its computational complexity is so large that a powerful GPU is required to guarantee the online availability. In [10], a semidense depth filtering formulation is proposed, which significantly reduces computational complexity. Thus, it is able to run online on a CPU and even on a smartphone [11]. With the assistance of key-points methods, direct tracking method even can achieve higher frame-rates on embedded platforms [12].

In both feature-based methods and direct methods, monocular SLAM can only get the reconstruction of scene up to a scale. There still exist more challenges of scale drift, which is seen as the major reason for accumulated error [13]. Due to the scale-drift phenomenon in monocular SLAM, the 3D similarity transform is adopted to represent camera poses instead of rigid body transformations [12]. And also loop closures detection is used as constraints to correct the scale drift. Using landmarks is also a method to reduce the scale drift, but it only can be applied in a cooperative scene [14]. As to the scale-ambivalent for the monocular SLAM method, it must be handled with either scale as a dubious factor or some additional information, such as point coordinates on a calibration object in the world coordinate and must be introduced to calculate the scale [15], and exploiting nonhomonymic motion constraints also work [16]. In our work, we solve the two problems by using a chessboard as a calibration reference. We need to be reminded here that the chessboard is widely and easily used. Therefore, in our approach, the chessboard is supposed to be involved in the visual field at the beginning of the SLAM process.

Contributions of This Paper. We present a metric online direct semidense monocular SLAM method, which calculates the real-time camera pose and builds consistent maps of the environment in metric scale, even in a large-scale environment. The method estimates the depth maps using a filtering-based algorithm, coupled with direct image alignment, and a chessboard is used to provide an initial depth estimation, in which the scale of the world is introduced into the mapping. The method presents steady tracking of the motion of the camera and accurate mapping of the environment and can run in real time on a CPU. The main contributions of this paper are (1) a method to introduce the metric scale into the monocular SLAM system and (2) a method to correct scale drift with a calibration object where a rule is proposed to determine when the calibration object is within the horizon of the camera and required to be detected for the purpose of reducing computation cost.

2. Preliminaries

In this section, some relevant mathematical definitions and notations used in this paper will be introduced. In particular, we introduce the most widely used camera model, pinhole camera model (Section 2.1), and represent the 3D poses as elements of Lie-Algebras (Section 2.2). In addition, we briefly introduce the solution for a weighted least-squares minimization on Lie-manifolds (Section 2.3).

Notations. Matrices are donated by bold, capital letters (R) and vectors by bold, lower case letters (ξ):

$\mathbf{R}_{ab}$: direction cosine matrix between a- and b-coordinates;

$\mathbf{t}_{ab}$: relative translation-vector between a- and b-coordinates;

$\mathbf{K}_i$: camera matrix for ith camera;

$\mathbf{X}_C, \mathbf{X}_W$: 3D point coordinate in camera coordinate or world coordinate;

Ω: set of normalized pixel coordinates;

$I : \Omega \to R$: images;

$D : \Omega \to R^+$: inverse depth map;

$V : \Omega \to R^+$: inverse depth variance map.

2.1. Camera Model and Coordinate Definition. The most widely used camera model is the pinhole camera model which is shown in Figure 1.

As described in Figure 1, the camera perspective model can be expressed as

$$\mathbf{Z}_c \cdot \mathbf{r}_{O_I P_I} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{r}_{O_C P_C} \\ 1 \end{bmatrix} = \begin{bmatrix} a_x & 0 & u_0 & 0 \\ 0 & a_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{WC} & \mathbf{t}_{WC} \\ 0_{(1\times 3)} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}_{O_W P_W} \\ 1 \end{bmatrix}, \tag{1}$$

where (u_0, v_0) is the camera principal point, f is the metric focal length, and dx and dy denote the physical width and height of one pixel. $\mathbf{R}_{WC}$, $\mathbf{t}_{WC}$ are external coefficients between C and W.

As described in aforementioned introduction, a chessboard is used to initialize the depth map. During the initialization process, the relationship between the camera coordinate and the world coordinate is estimated. Herein, the world coordinate is defined as follows: the original point is the left-top point of the chessboard, the X-coordinate is the top line of the chessboard and points to the right, and Y-coordinate is the left line of the chessboard and points to the bottom, as described in Figure 2.

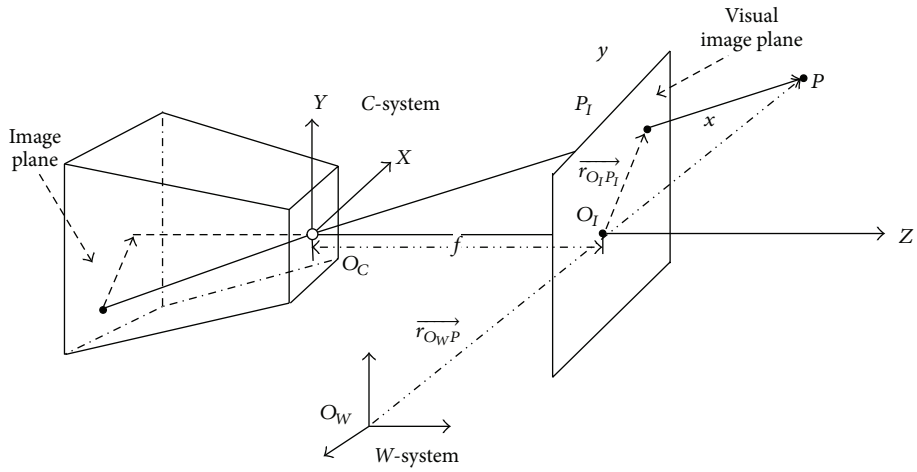

Figure 1: Camera perspective model.

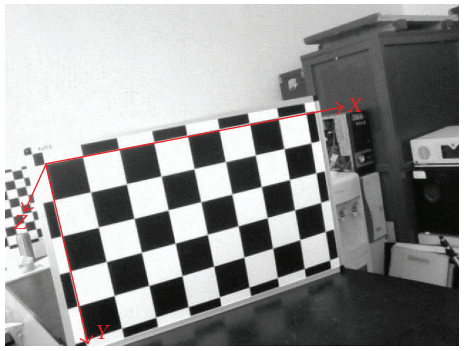

Figure 2: World coordinate definition.

2.2. 3D Pose Representing as Elements of Lie-Algebras. In this section,the representation of 3D pose transformation is described in the same way as in previous researches, such as [9, 10]. In order to guarantee the fluency of the presentation, we still utilize the most commonly used description of this problem. Usually, 3D rigid body transform $\mathbf{G} \in \mathrm{SE}(3)$ denotes translation and rotation in 3D that is defined by

$$\mathbf{G} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix}, \quad \text{with } \mathbf{R} \in \mathrm{SO}(3),\ \mathbf{t} \in R^3. \tag{2}$$

And 3D similarity transform $\mathbf{S} \in \mathrm{Sim}(3)$ denotes scaling, translation, and rotation that is defined by

$$\mathbf{G} = \begin{pmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix}, \quad \text{with } \mathbf{R} \in \mathrm{SO}(3),\ \mathbf{t} \in R^3,\ s \in R^+. \tag{3}$$

But a nonredundant expression for the camera pose is needed during optimization, which cannot be given by definition above, so the corresponding element $\boldsymbol{\xi} \in \mathrm{SE}(3)$ of the associated Lie-algebra is used to represent 3D rigid body transform and $\boldsymbol{\xi} \in \mathrm{Sim}(3)$ for 3D similarity transform. Elements are transformed into SE(3) by the exponential map $\mathbf{G} = \exp_{\mathrm{SE}(3)}(\boldsymbol{\xi})$ for rigid body transform and into Sim(3) by map $\mathbf{G} = \exp_{\mathrm{Sim}(3)}(\xi)$ for similarity transform, and their inverse is denoted by $\boldsymbol{\xi} = \log_{\mathrm{SE}(3)}(\mathbf{G})$ and $\boldsymbol{\xi} = \log_{\mathrm{Sim}(3)}(\mathbf{G})$. So the transformation moving a point from frame i to frame j is written as $\boldsymbol{\xi}_{ij}$. As in rigid body transform description, the pose concatenation operator $\circ : \mathrm{SE}(3) \times \mathrm{SE}(3) \to \mathrm{SE}(3)$ is defined as

$$\begin{aligned} \boldsymbol{\xi}_{ki} &:= \boldsymbol{\xi}_{kj} \circ \boldsymbol{\xi}_{ji} \\ &:= \log_{\mathrm{SE}(3)} \left(\exp_{\mathrm{SE}(3)} \left(\boldsymbol{\xi}_{kj} \right) \cdot \exp_{\mathrm{SE}(3)} \left(\boldsymbol{\xi}_{ji} \right) \right) \end{aligned} \tag{4}$$

which can be defined analogously for similarity transform. Please see [10] for more details.

2.3. Solution for Weighted Gauss-Newton Optimization. The Gauss-Newton algorithm is effective with nonlinear least-squares problems, with the advantage of a small computation cost [17]. The problem is usually described as follows: given m functions $\mathbf{r} = (r_1, \ldots, r_m)$ of n variables $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n)$, with $m > n$, the Gauss-Newton algorithm iteratively finds the minimum of the sum of squares:

$$S(\boldsymbol{\beta}) = \sum_{i=1}^{m} r_i^2(\boldsymbol{\beta}). \tag{5}$$

Beginning with an initial guess $\boldsymbol{\beta}^{(0)}$, the method proceeds with the iterations [17]:

$$\boldsymbol{\beta}^{(s+1)} = \boldsymbol{\beta}^{(s)} - \left(\mathbf{J}_{\mathbf{r}}^T \mathbf{J}_{\mathbf{r}} \right)^{-1} \mathbf{J}_{\mathbf{r}}^T \mathbf{r} \left(\boldsymbol{\beta}^{(s)} \right), \tag{6}$$

where if $\mathbf{r}$ and $\boldsymbol{\beta}$ are column vectors, the entries of the Jacobian matrix are [17]

$$\left(\mathbf{J}_{\mathbf{r}} \right)_{ij} = \frac{\partial r_i \left(\boldsymbol{\beta}^{(s)} \right)}{\partial \beta_j}. \tag{7}$$

And an iteratively reweighted least-squares problem is proposed to be robust to outliers arising, for example, from occlusions or reflections, which can be expressed as [17]

$$S(\boldsymbol{\beta}) = \sum_{i=1}^{m} \omega_i(\boldsymbol{\xi}) r_i^2(\boldsymbol{\beta}) \tag{8}$$

and can be proceeded by the iterations [17]

$$\boldsymbol{\beta}^{(s+1)} = \boldsymbol{\beta}^{(s)} - \left(\mathbf{J}_{\mathbf{r}}^{T}\mathbf{W}\mathbf{J}_{\mathbf{r}}\right)^{-1}\mathbf{J}_{\mathbf{r}}^{T}\mathbf{W}\mathbf{r}\left(\boldsymbol{\beta}^{(s)}\right), \tag{9}$$

where $\mathbf{W} = \mathbf{W}(\boldsymbol{\beta}^{(s)})$ is a weight matrix computed in each iteration and used to downweight large residuals.

3. Metric Online Monocular SLAM

This main contribution of this paper is that it provides a metric online monocular SLAM approach, by using a commonly used chessboard reference. The process can be divided into 5 parts. The chessboard provides the initial depth estimation of the scene, and the initial guess can be seen as the scaled source of the scene reconstruction. The initial depth estimation results are able to correct the scale drift through the key frames transfer, and thus we can obtain a global metric reconstruction. The whole process of this approach is shown in Figure 3.

In this section, we introduce our work from 4 parts: the initial depth estimation in Section 3.1, the estimation of camera pose using alignment in Section 3.2, method to correct the accumulated pose error with the aid of a chessboard in Section 3.3, and the depth map estimation and optimization in Section 3.4.

3.1. Initial Metric Depth Estimation. In the initial process, unlike in the traditional LSD-SLAM approach, we use a standard, key point-based method to obtain the initial depth map with the aid of a calibration object, which is a commonly used chessboard in this paper. We need to be reminded here that any other reference object is also good to fulfill this initial depth estimation process.

During the calibration process, the chessboard corners detection should be executed at the beginning. With the known 2D coordinates of chessboard corners and corresponding 3D coordinates, the relative pose of camera to the world coordinate can be got, which is known as the PNP problem [18].

3.1.1. Image Points Matching. To run online, only the depth of pixels with sufficiently large intensity gradient, which means that the pixel is a corner or on the edges, is estimated. We search the corresponding points of those pixels on the epipolar lines in the second image using a window-based matching approach with a window size of 3 pixels. Also, parallax constraint and sequence constraint are used to reduce the mismatching.

3.1.2. Initial Depth Map Estimation. With the known intrinsic camera parameters, extrinsic parameters, and corresponding image point pairs, the initial depth map can be estimated:

$$\begin{bmatrix}\mathbf{R}_{c_1 w}^{-1}\mathbf{K}^{-1}\mathbf{x}_1 & -\mathbf{R}_{c_2 w}^{-1}\mathbf{K}^{-1}\mathbf{x}_2\end{bmatrix}\begin{bmatrix}k_1\\k_2\end{bmatrix} = \mathbf{R}_{c_1 w}^{-1}\mathbf{t}_{c_1}^{w} - \mathbf{R}_{c_2 w}^{-1}\mathbf{t}_{c_2}^{w}, \tag{10}$$

where k_1, k_2 is inverse depth of pixel in the 1st image and 2nd image and $\mathbf{x}_1, \mathbf{x}_2$ is the homogeneous image pixel coordinate.

3.2. Pose Tracking. As a recursive process in the visual localization, the pose tracking is the main task. Now the commonly used pose tracking method is the image alignment, in which image sequences sampled in different time steps are consequently utilized, to provide the location verification of the moving camera. As the same in traditional monocular visual odometry researches, such as [7], we use the image alignment method by utilizing the direct features. The 3D pose $\boldsymbol{\xi}_{ji} \in \mathrm{SE}(3)$ of a new frame related to its keyframe is calculated using the direct SE(3) image alignment, and pose of keyframe is tracked using the direct Sim(3) image alignment. We need to be reminded here that, as described in the commonly used approaches, we use the same image alignment based pose tracking approach; please see [19] for more details. In order to make the description more fluent, we use the same nomination during the problem formulation.

3.2.1. Direct SE(3) *Image Alignment.* The pose estimation of a new frame is treated as a problem to minimize the variance-normalized photometric error:

$$E_p\left(\boldsymbol{\xi}_{ji}\right) = \sum_{\mathbf{p}\in\Omega_{D_i}}\left\|\frac{r_p^2\left(\mathbf{p},\boldsymbol{\xi}_{ji}\right)}{\sigma^2_{r_p(\mathbf{p},\boldsymbol{\xi}_{ji})}}\right\|_{\delta}$$
$$\text{with } r_p^2\left(\mathbf{p},\boldsymbol{\xi}_{ji}\right) \coloneqq I_i\left(\mathbf{p}\right) - I_j\left(\omega\left(\mathbf{p}, D_i\left(\mathbf{p}\right), \boldsymbol{\xi}_{ji}\right)\right),\ \sigma^2_{r_p(\mathbf{p},\boldsymbol{\xi}_{ji})} \coloneqq 2\sigma_I^2 + \left(\frac{\partial r_p^2\left(\mathbf{p},\boldsymbol{\xi}_{ji}\right)}{\partial D_i\left(\mathbf{p}\right)}\right)^2 V_i\left(\mathbf{p}\right), \tag{11}$$

where $\|\cdot\|_{\delta}$ is the Huber norm

$$\left\|r^2\right\|_{\delta} \coloneqq \begin{cases}\dfrac{r^2}{2\delta} & \text{if } |r| \le \delta\\ |r| - \dfrac{\delta}{2} & \text{otherwise.}\end{cases} \tag{12}$$

The Huber norm is applied to normalize the estimation residual. The residual's variance $\sigma^2_{r_p(\mathbf{p},\boldsymbol{\xi}_{ji})}$ is computed using covariance propagation:

$$\sum_f \approx \mathbf{J}_f \sum_{\mathbf{x}} \mathbf{J}_f^{T}. \tag{13}$$

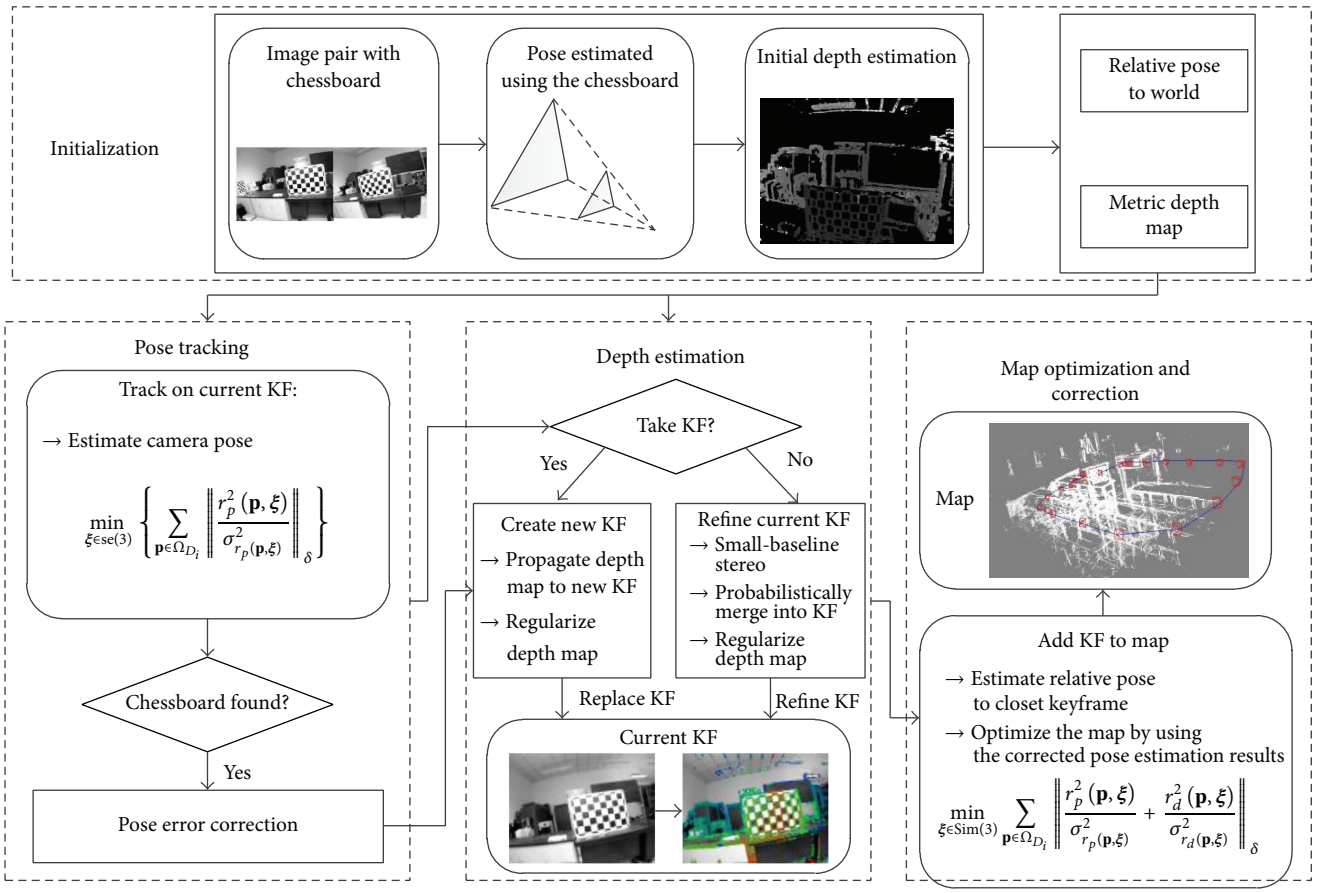

Figure 3: Overview over the whole SLAM system.

V_i is the inverse depth variance, and σ_I^2 is the image dense noise which is assumed to follow Gaussian distribution. The problem is solved using iteratively reweighted Gauss-Newton optimization described in Section 2.3.

3.2.2. Direct Sim(3) *Image Alignment.* To solve the scale-drift problem, direct Sim(3) image alignment is used to estimate edges between keyframes. After the depth map of a keyframe is refined, it is scaled to make its mean inverse depth to be one. Then, the direct Sim(3) image alignment is performed to elegantly incorporate the scaling difference between keyframes. Similarly to direct SE(3) image alignment, the pose between two keyframes, represented as 3D similarity transform $\mathbf{S} \in \mathrm{Sim}(3)$, is estimated as a problem to minimize the variance-normalized photometric error:

$$E\left(\boldsymbol{\xi}_{ji}\right) = \sum_{\mathbf{p}\in\Omega_{D_i}} \left\| \frac{r_p^2\left(\mathbf{p}, \boldsymbol{\xi}_{ji}\right)}{\sigma^2_{r_p(\mathbf{p},\boldsymbol{\xi}_{ji})}} + \frac{r_d^2\left(\mathbf{p}, \boldsymbol{\xi}_{ji}\right)}{\sigma^2_{r_d(\mathbf{p},\boldsymbol{\xi}_{ji})}} \right\|_\delta. \tag{14}$$

Here, a depth residual r_d is incorporated, which penalizes deviations in inverse depth between keyframes, allowing to directly estimate the scaled transformation between them. And the depth residual r_d and its variance $\sigma^2_{r_d(\mathbf{p},\boldsymbol{\xi}_{ji})}$ are computed as [10]

$$\begin{aligned} r_d\left(\mathbf{p}, \boldsymbol{\xi}_{ji}\right) &\coloneqq \left[\mathbf{p}'\right]_3 - D_j\left(\left[\mathbf{p}'\right]_{1,2}\right), \\ \sigma^2_{r_d(\mathbf{p},\boldsymbol{\xi}_{ji})} &\coloneqq \left(V_j\left(\left[\mathbf{p}'\right]_{1,2}\right)\right)\left(\frac{\partial r_d\left(\mathbf{p}, \boldsymbol{\xi}_{ji}\right)}{\partial D_j\left[\mathbf{p}'\right]_{1,2}}\right)^2 \\ &\quad + V_i\left(\mathbf{p}\right)\left(\frac{\partial r_d\left(\mathbf{p}, \boldsymbol{\xi}_{ji}\right)}{\partial D_i\left(\mathbf{p}\right)}\right)^2 \end{aligned} \tag{15}$$

and $\mathbf{p}' \coloneqq \omega_s(\mathbf{p}, D_i(\mathbf{p}), \boldsymbol{\xi}_{ji})$ denotes the corresponding point.

The problem can also be solved with iteratively reweighted Gauss-Newton optimization, which is the most commonly adopted approach in nonlinear optimization problem, as described in Section 2.3.

3.3. Pose Error Correction. In this section, we will show the outer assistance of a chessboard work in the pose error correction and in the depth maps accumulation errors correction. This is the main contribution of our work. Our SLAM approach is designed for the indoor robots, which means that it is possible for the robots to see the calibration object more

than once while moving. So it is practical to correct the pose estimation error accumulated with the calibration object.

3.3.1. Chessboard Corners Detection. It is unnecessary to detect the chessboard in every frame, which is of great computational cost. We give a principle to judge whether to detect the chessboard with the aid of pose of current frame.

When the four corners of the chessboard can be observed, the whole chessboard is inside the horizon, which can be used to decide when to detect the chessboard. In the world coordinate, the four corners of the chessboard are $\mathbf{X_1} = (0,0,0)$, $\mathbf{X}_2 = (w,0,0)$, $\mathbf{X}_3 = (0,h,0)$, and $\mathbf{X}_4 = (w,h,0)$, where w and h are the width and height of the chessboard. The pose of current frame to world coordinate can be calculated:

$$\begin{aligned}\mathbf{R}_{c_{\mathrm{cf}}}^{w} &= s\mathbf{R}_{c_{\mathrm{kf}}}^{w}\mathbf{R}_{c_{\mathrm{cf}}}^{c_{\mathrm{kf}}},\\ \mathbf{t}_{c_{\mathrm{cf}}}^{w} &= s\mathbf{R}_{c_{\mathrm{kf}}}^{w}\mathbf{t}_{c_{\mathrm{cf}}}^{c_{\mathrm{kf}}} + \mathbf{t}_{c_{\mathrm{kf}}}^{w},\end{aligned} \tag{16}$$

where $s\mathbf{R}_{c_{\mathrm{kf}}}^{w}$ and $\mathbf{t}_{c_{\mathrm{kf}}}^{w}$ are the pose of keyframe and $\mathbf{R}_{c_{\mathrm{cf}}}^{c_{\mathrm{kf}}}$ and $\mathbf{t}_{c_{\mathrm{cf}}}^{c_{\mathrm{kf}}}$ are the relative pose of current frame to current keyframe.

And homogeneous image pixel coordinate can be expressed as

$$\mathbf{x}_i = \mathbf{K}\left(\mathbf{R}_{w}^{c_{\mathrm{cf}}}\mathbf{X}_i + \mathbf{t}_{w}^{c_{\mathrm{kf}}}\right), \quad i = 1,2,3,4. \tag{17}$$

A chessboard detection is performed only under the condition that $\mathbf{x}_i$ is within the image:

$$\begin{aligned}0 \le \frac{\mathbf{x}_{i(1,1)}}{\mathbf{x}_{i(3,1)}} \le L_w,\\ 0 \le \frac{\mathbf{x}_{i(2,1)}}{\mathbf{x}_{i(3,1)}} \le L_h,\\ i = 1,2,3,4.\end{aligned} \tag{18}$$

Usually, successful chessboard detection is hard to achieve when the camera is too far away from the chessboard, even when the whole chessboard can be observed, so we add a limitation to avoid the case:

$$\left\|\mathbf{t}_{c_{\mathrm{cf}}}^{w}\right\| \le 3. \tag{19}$$

3.3.2. Pose Correction. When chessboard detection is performed successfully on current frame while failing on previous frame, a new keyframe will be created with a corrected pose which is estimated with the correspondence between the image coordinate and the world coordinate of those chessboard corners. Then, relative pose between new created keyframe and previous keyframe can be calculated as

$$\begin{aligned}\mathbf{R}_{c_{\mathrm{cf}}}^{c_{\mathrm{kf}}} &= \frac{1}{s}\mathbf{R}_{c_{\mathrm{kf}}}^{w\ -1}\mathbf{R}_{c_{\mathrm{cf}}}^{w},\\ \mathbf{t}_{c_{\mathrm{cf}}}^{c_{\mathrm{kf}}} &= \frac{1}{s}\mathbf{R}_{c_{\mathrm{kf}}}^{w}\left(\mathbf{t}_{c_{\mathrm{kf}}}^{w} - \mathbf{t}_{c_{\mathrm{cf}}}^{w}\right)\end{aligned} \tag{20}$$

with which the depth map of new keyframe can be initialized by projecting points from the previous frame.

3.4. Depth Map Estimation and Optimization. The depth map estimation problem is the most commonly referred problem in monocular SLAM, and in this paper we still use the most common method to execute the depth map estimation using the method proposed in [19]. When a new frame is obtained, we first measure whether the camera has moved far away enough from its keyframe that a new keyframe should be created using it. To do this, a weighted combination of relative distance and angle to the current keyframe is threshold, which is the same as in [19].

After the pose estimation process, the pose graph optimization is necessary to continuously optimize the map which consists of a set of keyframes and their camera poses. The error function is defined in the following equation, the same in [20],

$$\begin{aligned}&E\left(\boldsymbol{\xi}_{W1}\cdots\boldsymbol{\xi}_{Wn}\right)\\ &\quad\coloneqq \sum_{(\xi_{ji},\Phi_{ji})\in\varepsilon}\left(\boldsymbol{\xi}_{ji}\cdot\boldsymbol{\xi}_{Wi}^{-1}\cdot\boldsymbol{\xi}_{Wj}\right)^{T}\Phi_{ji}^{-1}\left(\boldsymbol{\xi}_{ji}\cdot\boldsymbol{\xi}_{Wi}^{-1}\cdot\boldsymbol{\xi}_{Wj}\right).\end{aligned} \tag{21}$$

4. Experiments

In the experiments process, the SLAM approach is executed in an indoor environment, where the visual scene is occupied by artificial equipment. In the experiment, the camera is selected as a commonly used industrial camera, the frame-frequency of which is higher than 30 frames/sec.

Before the experiment, the camera is placed in front of a chessboard for the initial alignment, from which the initial positions and gestures between camera and word coordinate are calculated. During the experiment process, the hand-held camera is moved around the house arbitrarily and returned to the start position at the end of the experiment. Based on the same datasets sampled in the moving process, our SLAM approach is run twice to provide the comparisons of these two different methods. The first experiment utilizes the most common monocular SLAM approach, as described in [9]. The second experiment is calculated with the assistance of the chessboard, especially in the correction of error accumulation. Both the reconstruction results and the ego-motion estimation results are depicted in Figure 4. Herein the SLAM results in only two keyframes are provided for simplicity, and the comparison in the whole moving process is the same as in the selected keyframes.

As depicted in the experimental results, we can easily find that there exits accumulated error in pose estimation when the traditional LSD-SLAM method is used, which is shown in Figure 4. As a result, the corresponding reconstruction results are also degenerated obviously. Thus, we can intuitively find that there exist ghost images of some reconstructed objects. While the proposed method with the outer reference assistance is used, the ego-motion estimation results can be improved, from which the reconstruction results can also be corrected with the assistance of chessboard calibration. Thus, we can achieve higher accuracy mapping of the experiment scene, and the reconstruction map of the scene in the second method is able to obtain solely results, without any ghost reconstruction in Figures 5 and 6.

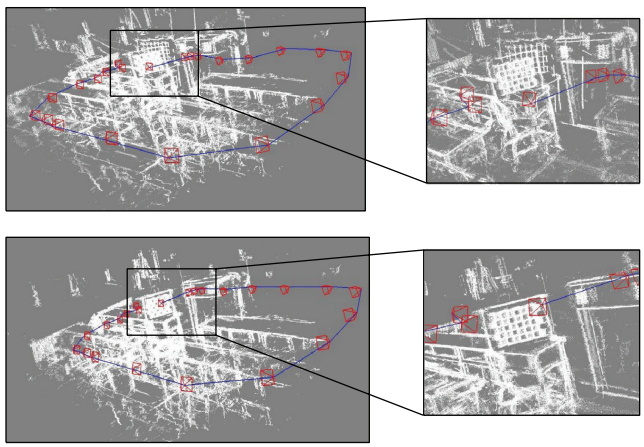

Figure 4: Reconstruction of scene.

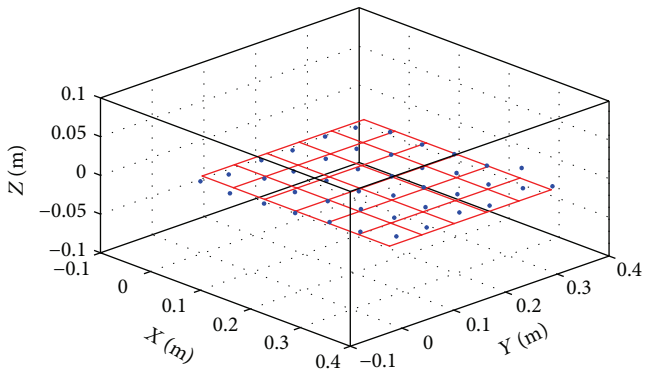

Figure 5: A comparison between the reconstructed points and the real ones. The red points are real chessboard corners and the blue ones are those reconstructed.

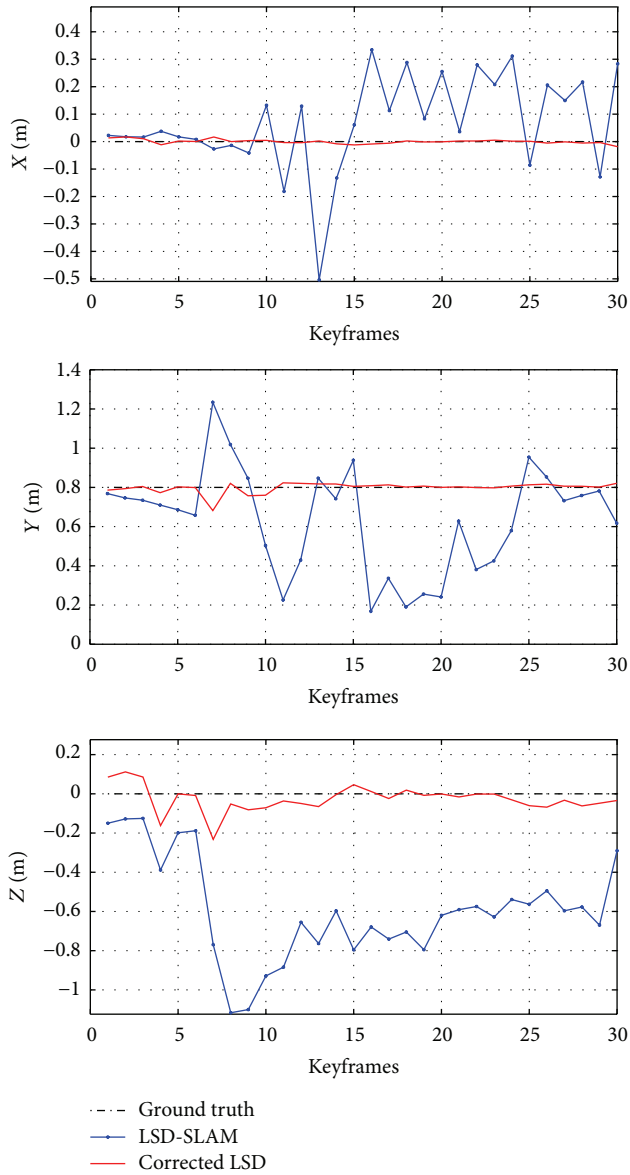

Figure 6: Reconstructed chessboard corners in different keyframes.

The aforementioned results can only provide a visualized comparison. In order to assess the proposed method quantitatively, we also provide a numerical analysis of the reconstruction results. Because the around scene during the experimental process is randomly selected, without any other information about the accurate scale and size, thus we choose the reconstructed result of the known chessboard to verify the accuracy of our method. As depicted in Figure 5, the comparison between the reconstructed chessboard corners and the real ones is shown.

As depicted in Figure 5, because our SLAM system can achieve a metric reconstruction of scene, we can find that the reconstruction precision is well improved remarkably with the assistance of the chessboard measurements.

In addition, all the coordinates of a randomly selected chessboard corner in different keyframes are also counted, and it is used to show the variation of accuracy about the reconstructed results.

From the results shown in Figure 6, we find that, at the beginning of the experiments, both the original LSD-SLAM method and the proposed corrected LSD-SLAM method can achieve accurate reconstruction of the chessboard corner. But along with the increasing measurements, the reconstruction error of the original LSD-SLAM method increases rapidly, while that of those proposed method is kept in a limit range, which indicates that our method can correct the accumulative error.

5. Conclusion

In this article, a metric direct monocular SLAM system is introduced, which can run in real time on a CPU and can obtain metric reconstruction of the scene. Based on the assistance of a chessboard, the initial depth map is estimated; meanwhile, the similarity transform between known world coordinate and the map coordinate is calculated, which can be used to convert the map to the known world coordinate. The system is tested in a complex indoor environment, and its accuracy is verified with a comparison between the estimated

chessboard corner coordinates and the real ones. The indoor experiments prove the effectiveness of the proposed metric monocular SLAM approach.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by the National Natural Science Fund of China 61403398.

References

[1] M.-H. Le, A. Vavilin, and K.-H. Jo, "3D scene reconstruction enhancement method based on automatic context analysis and convex optimization," *Neurocomputing*, vol. 137, pp. 71–78, 2014.

[2] H. Yu, R. Sharma, R. W. Beard, and C. N. Taylor, "Observability-based local path planning and obstacle avoidance using bearing-only measurements," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1392–1405, 2013.

[3] D. Yang, Z. Liu, F. Sun, J. Zhang, H. Liu, and S. Wang, "Recursive depth parameterization of ego-motion estimating: observability analysis and performance evaluation," *Information Sciences*, vol. 287, pp. 38–49, 2014.

[4] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

[5] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *Proceedings of the 26th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '13)*, pp. 2100–2106, Tokyo, Japan, November 2013.

[6] M. Dhome, M. Richetin, J.-T. Lapreste, and G. Rives, "Determination of the attitude of 3-D objects from a single perspective view," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 12, pp. 1265–1278, 1989.

[7] J. Stühmer, S. Gumhold, and D. Cremers, "Real-time dense geometry from a handheld camera," in *Pattern Recognition*, M. Goesele, S. Roth, A. Kuijper, B. Schiele, and K. Schindler, Eds., vol. 6376 of *Lecture Notes in Computer Science*, pp. 11–20, Springer, Berlin, Germany, 2010.

[8] 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.

[9] D. Yang, F. Sun, S. Wang, and J. Zhang, "Simultaneous estimation of ego-motion and vehicle distance by using a monocular camera," *Science China Information Sciences*, vol. 57, no. 5, pp. 1–10, 2014.

[10] J. Engel, J. Sturm, and D. Cremers, "Scale-aware navigation of a low-cost quadrocopter with a monocular camera," *Robotics and Autonomous Systems*, vol. 62, no. 11, pp. 1646–1656, 2014.

[11] T. Schops, J. Enge, and D. Cremers, "Semi-dense visual odometry for AR on a smartphone," in *Proceedings of the 13th IEEE International Symposium on Mixed and Augmented Reality (ISMAR '14)*, pp. 145–150, IEEE, Munich, Germany, September 2014.

[12] E. Rodolà, A. Albarelli, D. Cremers, and A. Torsello, "A simple and effective relevance-based point sampling for 3D shapes," *Pattern Recognition Letters*, vol. 59, pp. 41–47, 2015.

[13] H. Strasdat, J. Montiel, and A. Davison, "Scale drift-aware large scale monocular SLAM," *Robotics: Science and Systems*, vol. 6, no. 6, pp. 56–62, 2010.

[14] E. Hernández, J. M. Ibarra, J. Neira, and R. Cisneros, "Visual SLAM with oriented landmarks and partial odometry," in *Proceedings of the 21st International Conference on Electrical Communications and Computers*, pp. 39–45, San Andres Cholula, Mexico, September 2011.

[15] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proceedings of the International Conference on Computer Vision (ICCV '03)*, pp. 1403–1410, Nice, France, October 2003.

[16] N. Ohnishi and A. Imiya, "Appearance-based navigation and homing for autonomous mobile robot," *Image and Vision Computing*, vol. 31, no. 6-7, pp. 511–532, 2013.

[17] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, Pa, USA, 1996.

[18] Y. Wu and Z. Hu, "PnP problem revisited," *Journal of Mathematical Imaging and Vision*, vol. 24, no. 1, pp. 131–141, 2006.

[19] J. Engel, T. Schops, and D. Cremers, "LSD-SLAM: large-scale direct monocular SLAM," in *Proceedings of the European Conference on Computer Vision (ECCV '14)*, pp. 834–849, Zurich, Switzerland, September 2014.

[20] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: a general framework for graph optimization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '11)*, pp. 3607–3613, Shanghai, China, May 2011.