

Research Article

Pose Estimation of a Noncooperative Target Based on Monocular Visual SLAM

Ting Lei ¹, Xiao-Feng Liu ¹, Guo-Ping Cai ¹, Yun-Meng Liu,² and Pan Liu²

¹Department of Engineering Mechanics, State Key Laboratory of Ocean Engineering, Shanghai Jiaotong University, Shanghai 200240, China

²Shanghai Institute of Technical Physics of the Chinese Academy of Science, Shanghai 200083, China

Correspondence should be addressed to Guo-Ping Cai; caigp@sjtu.edu.cn

Received 26 July 2019; Revised 20 October 2019; Accepted 25 October 2019; Published 7 December 2019

Academic Editor: Paolo Tortora

Copyright © 2019 Ting Lei et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper estimates the pose of a noncooperative space target utilizing a direct method of monocular visual simultaneous location and mapping (SLAM). A Large Scale Direct SLAM (LSD-SLAM) algorithm for pose estimation based on photometric residual of pixel intensities is provided to overcome the limitation of existing feature-based on-orbit pose estimation methods. Firstly, new sequence images of the on-orbit target are continuously inputted, and the pose of each current frame is calculated according to minimizing the photometric residual of pixel intensities. Secondly, frames are distinguished as keyframes or normal frames according to the pose relationship, and these frames are used to optimize the local map points. After that, the optimized local map points are added to the back-end map. Finally, the poses of keyframes are further enumerated and optimized in the back-end thread based on the map points and the photometric residual between the keyframes. Numerical simulations and experiments are carried out to prove the validity of the proposed algorithm, and the results elucidate the effectiveness of the algorithm in estimating the pose of the noncooperative target.

1. Introduction

With the development of space technology, on-orbit service is getting widespread attentions. And on-orbit capture of space targets is the key of it, which is regarded as the primary issue to be addressed by many space missions such as the assemblage of space station, fuel filling, or the maintenance of spacecraft. On-orbit capture means the operation that uses service spacecraft to capture the target spacecraft. The space targets can be divided into cooperative targets and noncooperative targets. Cooperative targets can provide kinematics and dynamics information, such as velocity, pose, mass, inertia, centroid position, and size, to facilitate the subsequent design of the capturing path and control law, whereas noncooperative targets cannot. How to capture a noncooperative target under the condition of less effective information is becoming one of the focuses in the aerospace industry. In recent years, the progress in visual technology has provided new ideas and approaches for motion observation, motion prediction, and three-dimensional structural reconstruction

of noncooperative targets. Vision-based target observation technology adopts a camera as the sensor, which has the characteristics of low cost, easy to install, and noncontact. It is very suitable for on-orbit service and other missions of spacecraft. Therefore, the researches on the visual technology of spacecraft are of great scientific significance and engineering application value.

At present, monocular pose estimation has been applied to many missions of a spacecraft, such as autonomous navigation [1–5] and rendezvous operation [6, 7]. The relative pose measurement and estimation of service spacecraft and space target is also the important premise of on-orbit capture. Up to now, there are some researches on it and some achievements have been made. For example, Wen et al. [8] extracted circles, lines, and points on the cooperative targets and calculated the target pose by the Point-3-Perspective algorithm. The algorithm is suitable for real-time visual measurement that requires high precision in aerospace, which is not suitable for noncooperative targets. Song and Cao [9] proposed a monocular vision pose measurement method

based on solar triangle structure. The algorithm recognizes feature structure based on sliding window Hough transformation and proposed inscribed circle of a triangle, calculating the relative pose of target expressed by rotation and translation matrix. Regoli et al. [10] presented an approach for estimating the pose of an unknown object using Photonic Mixer Device cameras. The algorithm works online by making use of the amplitude and depth information provided by the camera at high frame rates to estimate the relative pose in the rendezvous and docking process. D'Amico et al. [11] used a known three-dimensional model of a passive space resident object and single low-resolution two-dimensional images collected by the active spacecraft to estimate the pose of a noncooperative target, but the method had the disadvantage of requiring lots of known information of the target. Li et al. [12] proposed a relative pose estimation method between noncooperative spacecrafts based on parallel binocular vision. The method extracted the line features and feature points on the freely tumbling target and calculated the relative pose between the target coordinate and the world coordinate. Shtark and Gurfil [13] developed a computer-vision feature detection and matching algorithm to identify and locate the target in the captured images and then designed three different filters to estimate the relative position and velocity. He et al. [14] proposed a measurement method of relative position and attitude between two noncooperative spacecrafts based on graph cut and edge information algorithm. The circular feature of the target is accurately extracted and the edges are ellipse fitting, and the relative position and attitude of target is obtained by fitting ellipse parameters of binocular cameras. Dong and Zhu [15] developed a real-time vision-based pose and motion estimation of noncooperative target by an extended Kalman filter. Optical flow algorithm was adopted to track the feature points of the target, and photogrammetry was used to provide more accurate initial conditions. Mortari et al. [16] calculated the centroid and distance from the observer to the body by an image processing approach of illuminated ellipsoid and estimated the observer-to-body relative position in inertial coordinates for navigation purposes. Modenini [17] utilized some obtained analytical results for the perspective projection of an ellipsoid and simplified the attitude determination problem to an approximate orthogonal Procrustes problem, which greatly reduced the difficulty of the problem. Liu and Hu [18] developed a novel framework to determine the relative pose and range of a known-shaped noncooperative spacecraft from a single image, and the method was validated by synthetic and real images. Zhang et al. [19] addressed the problem of estimating the relative pose of a target spacecraft by employing Gaussian process regression. Experiments on a simulated image dataset that contains satellite images of 1D and 2D pose variation were performed, and the results validated the effectiveness and robustness of the approach. It is worth pointing out here that most strategies of the exiting studies on relative pose measurement and estimation of space target are matching the target with the existing template library, or extracting the features on the target such as corners, circular dockings, rectangular sails, and other features, which is the basis of the calculation. Matching and

calculating based on the template library requires the three-dimensional structure information of the target in advance. Establishing a library containing a large number of representative satellite models consumes a lot of resources and time; thus, it is not desirable in actual operation. And calculating the relative pose based on target features requires extraction of strong geometric features on the target. When there are no such features on the target or the local textures are missing, the accuracy of the calculation will be greatly reduced. Therefore, the problem that the existing algorithms are not quite robust to weak textures still exists.

The visual SLAM method was originally proposed to solve the localization problem of a mobile robot [20, 21]. It uses a camera as the unique sensor to track the pose of the camera in real time according to the sequence images taken by the camera and at the same time constructs the three-dimensional map of the environment. In the visual SLAM method, the scene is stationary, and the camera is moving relative to the scene. When using a camera to estimate the relative pose of a noncooperative space target, the camera fixed to the service spacecraft is stationary and the space target is moving relative to the service spacecraft. Therefore, it is reasonable to apply the SLAM method to the relative pose estimation of noncooperative space targets. Tweddle [22] described a new approach to solving a SLAM problem for uncooperative objects that are spinning about an arbitrary axis. The method estimated a geometric map of the target and obtained its dynamic and kinematic parameters. Augenstein and Rock [23] presented an algorithm for real-time pose estimation using monocular SLAM/SFM by combining Bayesian estimation methods and measurement inversion techniques. The performance and viability of the hybrid approach were demonstrated by numerical simulations and field experiments. Chiodini et al. [24] presented a collaborative visual localization method for rovers which designed to hop and tumble across the surface of small Solar System bodies. By capturing images from various poses and illumination angles, the spacecraft mapped the surface of the body and created a prior 3D landmark map. Then, the hopping rover relocalized the prior map and performed simultaneous localization and mapping. The method was evaluated with image sequences of a mock asteroid and was shown to be robust to varying illumination angles, scene scale changes, and off-nadir camera pointing angles. Visual SLAM is mainly divided into feature method and direct method. The feature method extracts some representative points on the image such as corners and edges. These features remain stable after the movement of the camera. When the camera is moving, the feature points observed on two images of adjacent positions are matched, and two matched points form a matched point pair. The relative pose of the camera at two different positions is estimated according to the corresponding relationship of the matched point pairs. However, when the number of the observable features in the scene is less than a certain account, the feature-based SLAM will be invalid. And if there are many repeated textures in the scene, many mismatches will occur in the feature-based SLAM, resulting in the decrease of estimation accuracy. Research in the field of space has its particularity, such as poor lighting conditions

and less effective features, and the algorithm should satisfy both real-time and accuracy requirements. Because of the above limitations, the feature-based SLAM, ORB-SLAM, for example, cannot be initialized in all cases [25]. The PnP method, which is commonly used in pose measurement, is easy to fail in the calculation process, for the reason that it uses limited number of features and is sensitive to noise and mismatches [26]. The direct SLAM is establishing the photometric residual of pixels on images of two adjacent positions when the camera moves and minimizing the photometric residual to estimate the pose of the camera. Due to the direct use of image intensity, direct SLAM does not depend on the number of features in the scene, so it has strong robustness under complex conditions such as occlusion and weak texture scenes and is not easily affected by such factors. It also meets the requirements of real time and accuracy.

In this paper, the relative pose is estimated by the method of the direct SLAM. The algorithm calculates the photometric residual based on the intensity of pixels on the image and minimizes the photometric residual by solving the optimization problem to estimate the pose of the camera. The estimated pose of the camera can be translated into the pose of the target. The structure of this paper is as follows. Section 2 outlines and explains the basic principles of the direct method. Section 3 briefly describes the LSD-SLAM algorithm used in this paper. Section 4 designs and carries out numerical simulations for the algorithm in this paper and analyzes the results. Section 5 is a summary of the preceding contents and the conclusions obtained.

2. Fundamental Principles of the Direct Method

Feature-based visual SLAM method estimates the pose by matching the features. Unlike that, the direct SLAM estimates the pose by calculating and optimizing the photometric residual. For two pictures taken by the camera in different perspectives, the photometric residual refers to the difference between the intensity value of a pixel in the first image and the intensity value of its corresponding pixel in the second image through the relative pose relationship. In the first image, each pixel can find a corresponding pixel in the second image, and a photometric residual can be calculated. The direct method calculates the sum of all the photometric residuals, and the estimated pose is obtained by minimizing the sum.

When the scene is stationary, the relationship between pixels on the image i and their projection positions on the image j is shown in Figure 1. The plane rectangular coordinate systems $O_i-u_i v_i$ and $O_j-u_j v_j$ are the pixel coordinate systems, built on the image planes of i and j , respectively. The space rectangular coordinate systems $C_i-x_i y_i z_i$ and $C_j-x_j y_j z_j$ are the camera coordinate systems of the image i and the image j , respectively, where $x_i C_i y_i$ is parallel to $u_i O_i v_i$ and $x_j C_j y_j$ is parallel to $u_j O_j v_j$. $T_{ji} \in \mathbb{R}^{3 \times 4}$ is the transformation matrix between $C_i-x_i y_i z_i$ and $C_j-x_j y_j z_j$, which is expressed as Equation (3). $\mu_{ik} = (u_{ik}, v_{ik})^T \in \mathbb{R}^{2 \times 1}$ is the coordinate vector of the pixel projected on the image i (that is the k -th pixel) of the space point p_k in the coordinate system $O_i-u_i v_i$, where

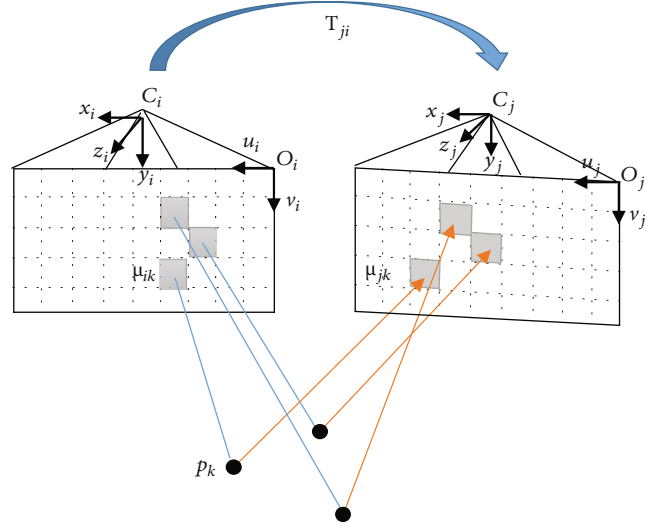


FIGURE 1: Schematic diagram of the relationship between the pixels on the image i and the image j and their projection positions.

T represents the transposition of a vector or a matrix. $\mu_{jk} = (u_{jk}, v_{jk})^T \in \mathbb{R}^{2 \times 1}$ is the coordinate vector of the pixel projected on the image j of the space point p_k in the coordinate system $O_j-u_j v_j$. The pixel is the corresponding projection pixel of the k -th pixel on the image i .

Suppose $\mathbf{x}_i = (x_{ik}, y_{ik}, z_{ik})^T$ and $\mathbf{x}_j = (x_{jk}, y_{jk}, z_{jk})^T$ are the coordinate vectors of the space point p_k in the coordinate systems $C_i-x_i y_i z_i$ and $C_j-x_j y_j z_j$, respectively. d_k is the depth of the k -th pixel on the image i , that is, the coordinate value of z coordinate in the coordinate system $C_i-x_i y_i z_i$ of the space point p_k [27]. We have the following expression

$$\mathbf{x}_i = (x_{ik}, y_{ik}, z_{ik})^T = d_k \mathbf{K}^{-1} (u_k, v_k, 1)^T, \quad (1)$$

where $\mathbf{K} \in \mathbb{R}^{3 \times 3}$ is the camera intrinsic matrix and $\mathbf{K}^{-1} \in \mathbb{R}^{3 \times 3}$ is its inverse matrix, given by

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (2)$$

$$\mathbf{K}^{-1} = \begin{bmatrix} f'_x & 0 & c'_x \\ 0 & f'_y & c'_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{f_x} & 0 & -\frac{c_x}{f_x} \\ 0 & \frac{1}{f_y} & -\frac{c_y}{f_y} \\ 0 & 0 & 1 \end{bmatrix},$$

where f_x and f_y are, respectively, the scaled focal lengths in the u and v directions on the image, $f'_x = (1/f_x)$ and $f'_y = (1/f_y)$. (c_x, c_y) is the pixel coordinate of the principle point of the camera, $c'_x = -(c_x/f_x)$ and $c'_y = -(c_y/f_y)$. The so-called principle point is the pixel coordinate of the intersection points of

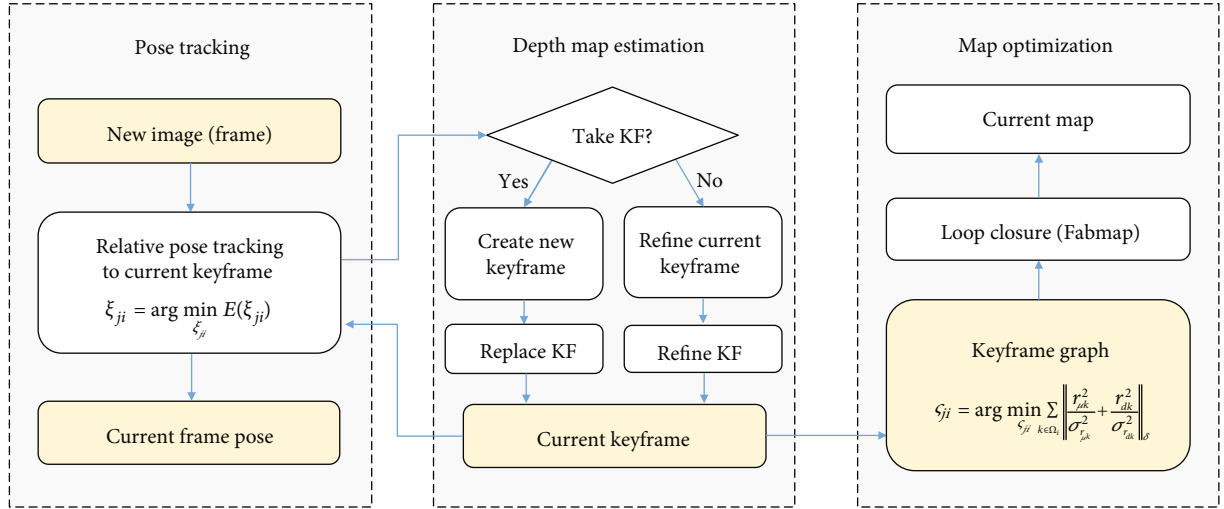


FIGURE 2: A broad outline of the LSD-SLAM algorithm.

C_i and C_j perpendicular to the respective image plane. For the camera center at different locations such as C_i and C_j , their c_x and c_y are constant.

The relative transformation matrix between the coordinate systems $C_i-x_iy_iz_i$ and $C_j-x_jy_jz_j$ can be expressed as [28]

$$\mathbf{T}_{ji} = [\mathbf{R}_{ji} \ \mathbf{t}_{ji}], \quad (3)$$

where $\mathbf{R}_{ji} \in \mathbb{R}^{3 \times 3}$ and $\mathbf{t}_{ji} \in \mathbb{R}^{3 \times 1}$ are the rotation matrix and translation vector from the coordinate system $C_i-x_iy_iz_i$ to the coordinate system $C_j-x_jy_jz_j$. Using Equation (3), the coordinates \mathbf{x}_j of the space point p_k in the camera coordinate system can be expressed as

$$\mathbf{x}_j = \begin{pmatrix} x_{jk} \\ y_{jk} \\ z_{jk} \end{pmatrix}^T = \mathbf{R}_{ji} \mathbf{x}_i + \mathbf{t}_{ji}. \quad (4)$$

The pixel coordinate $\mathbf{u}_{ik} = (u_{ik}, v_{ik})^T$ can be calculated, given by [29]

$$\begin{cases} u_{jk} = \left(\frac{x_{jk}}{z_{jk}} \right) f'_x + c'_x, \\ v_{jk} = \left(\frac{y_{jk}}{z_{jk}} \right) f'_y + c'_y. \end{cases} \quad (5)$$

Assume that I_{ik} and I_{jk} are the intensity value of the k -th pixel on the image i and its corresponding pixel on the image j , the photometric residual is given by

$$r_k = I_{ik}(\mathbf{u}_{ik}) - I_{jk}(\mathbf{u}_{jk}). \quad (6)$$

\mathbf{T}_{ji} can be obtained by solving the following optimization problem

$$\mathbf{T}_{ji} = \arg \min_{\mathbf{T}_{ji}} \sum_k \|r_k\|^2. \quad (7)$$

When the quadratic sum of all photometric residuals is minimized, the \mathbf{T}_{ji} that minimizes the sum is the estimated pose. This optimization problem can be solved by the gradient-based optimization methods such as the Gauss-Newton method or the Levenberg-Marquardt algorithm.

3. LSD-SLAM Algorithm

The LSD-SLAM utilizes the pixels whose gradient has significant changes and minimizes the photometric residual of pixels to estimate the pose, recover the depth, and construct the semidense three-dimensional map [30]. The process of LSD-SLAM is shown in Figure 2. There are three main modules of the algorithm: camera pose tracking, depth map estimation, and global map optimization. First, the pose tracking module continuously calculates the pose of the incoming frames. A single image in a long sequence of consecutive images is generally called a frame. Then, based on the obtained relative pose, the depth map estimation module generates and enhances keyframes in an independent thread; the construction and the refinement of the local 3D map of the scene are performed at the same time. Finally, the map optimization module filters all the keyframes and uses the optimization algorithm to perform more detail and accurate optimization of the global 3D map in another independent thread.

3.1. Camera Pose Tracking. When LSD-SLAM is running, the input frames are always the real-time images of the camera, or the images of a continuous sequence. The lately input frame is called the current frame. For each current frame j , assume that $\mathbf{T}_{j-1,i}$ is the relative pose transformation matrix

of the previous frame $j - 1$ relative to the nearest keyframe i . Then, take $\mathbf{T}_{j-1,i}$ as the initial value and minimize the normalized photometric residual between the frames i and j , the relative pose \mathbf{T}_{ji} of the current frame j to the nearest keyframe i is calculated, given by

$$E(\xi_{ji}) = \sum_{\mu_{ik} \in \Omega_i} \left\| \frac{r_{\mu_{ik}}^2}{\sigma_{r_{\mu_{ik}}}^2} \right\|_{\delta}, \quad (8)$$

$$r_{\mu_{ik}} = I_{ik}(\mathbf{u}_{ik}) - I_{jk}(\mathbf{u}_{jk}(\mathbf{u}_{ik}, d_{ik}, \xi_{ji})), \quad (9)$$

where E is the normalized photometric residual function; Ω_i is the set of all the pixels in the region where the pixel gradient has significant changes on the frame i . By considering geometric disparity error and photometric disparity error [31] comprehensively, it can be found that the greater the pixel gradient is, the smaller the error is. Therefore selecting pixels of "significant change in gradient" is to eliminate the comprehensive error. μ_{ik} is the pixel coordinate of the k -th pixel on the keyframe i , while μ_{jk} is the corresponding pixel coordinate on the current frame j . d_{ik} is the depth of the k -th pixel on the keyframe i ; ξ_{ji} is the transformation matrix \mathbf{T}_{ji} between frame i and frame j represented by Lie Algebras [32]. Based on μ_{ik} , d_{ik} , and ξ_{ji} , the corresponding pixel coordinate μ_{jk} on the frame j is determined. $\|\cdot\|_{\delta}$ is the Huber norm, $r_{\mu_{ik}}$ is the photometric residual, and $\sigma_{r_{\mu_{ik}}}^2$ is the variance of the photometric residual $r_{\mu_{ik}}$. The optimization problem is

$$\xi_{ji} = \arg \min_{\xi_{ji}} E(\xi_{ji}). \quad (10)$$

When the normalized photometric residual function takes the minimum value, the obtained ξ_{ji} is the estimated relative transformation. This optimization problem can be solved by the reweighted Gauss-Newton method [30].

3.2. Depth Map Estimation. Based on the obtained relative pose of each frame, LSD-SLAM performs depth map estimation. In the calculation process, the algorithm selects some representative frames as keyframes, which are refined by other frames during tracking. These keyframes are used to build the local map and enhance it. First, the weighted distance $\text{dist}(\xi_{ji})$ of the current frame j relative to keyframe i is calculated. If the distance is greater than a certain threshold, the current frame is taken as a new keyframe. The threshold is a variable value that depends on the number of existing keyframes, which can be found in source code (https://github.com/tum-vision/lsd_slam). The weighted distance can be expressed as

$$\text{dist}(\xi_{ji}) = \xi_{ji}^T \mathbf{W} \xi_{ji}, \quad (11)$$

where \mathbf{W} is the weight matrix. When the current frame j becomes a new keyframe, it replaces the previous keyframe as the latest keyframe. The former keyframe was added into the keyframe graph and optimized in the back-end thread.

The depth map points of the previous keyframe are projected to the current frame according to the relative pose and are set as the initial depth map points of the current frame. And if the distance is less than the threshold, the current frame is not set as a keyframe. Based on the relative pose between the frame j and the nearest keyframe i , the depth d_j and the variance σ_j^2 of the current frame j can be obtained. And the depth d_i and the variance V_i are updated by using the extended Kalman filter [33].

$$\begin{cases} d'_i(\mathbf{u}_j) \leftarrow \frac{V_i(\mathbf{u}_j) d_j(\mathbf{u}_j) + \sigma_j^2 d_i(\mathbf{u}_j)}{V_i(\mathbf{u}_j) + \sigma_j^2}, \\ V'_i(\mathbf{u}_j) \leftarrow \frac{V_i(\mathbf{u}_j) \sigma_j^2}{V_i(\mathbf{u}_j) + \sigma_j^2}. \end{cases} \quad (12)$$

The updated depth map is integrated into the original depth map, which makes the depth map of the keyframe more complete and smooth.

3.3. Global Map Optimization. Due to the scale uncertainty of monocular vision, the depth of pixels cannot be accurately obtained. It can only be estimated by moving the camera to different positions, and the error accumulates after long-term movements of the camera, which leads to the inconsistency of the scale in the scene. In other words, the scene will change dramatically. However, the pose of each frame is calculated relative to the nearest keyframe, and frames which are relative to the same keyframe make up a set. The consistency of the scale for the frames in the same set can be maintained, while the consistency of different sets cannot. Therefore, for the keyframes added to the keyframe graph, LSD-SLAM solves the optimization problem by minimizing the joint function of the normalized photometric residual and the normalized depth of the scene. The relative pose between two keyframes i and j can be obtained, which can be expressed as

$$\zeta_{ji} = \arg \min_{\zeta_{ji}} \sum_{\mu_{ik} \in \Omega_i} \left\| \frac{r_{\mu_{ik}}^2}{\sigma_{r_{\mu_{ik}}}^2} + \frac{r_{dk}^2}{\sigma_{r_{dk}}^2} \right\|_{\delta}, \quad (13)$$

$$\begin{aligned} r_{\mu_{ik}} &= I_{ik}(\mathbf{u}_{ik}) - I_{jk}(\mathbf{u}_{jk}(\mathbf{u}_{ik}, d_{ik}, \zeta_{ji})), \\ r_{dk} &= z_{jk} - d_{jk}(\mathbf{u}_{jk}), \end{aligned} \quad (14)$$

where r_{dk} is the difference of the z coordinate value z_{jk} of the space point p_k in the camera coordinate of the frame j , and the depth d_{jk} of the pixel coordinate μ_{jk} of the corresponding pixel in the frame j . $\sigma_{r_{dk}}^2$ is the variance of the residual r_{dk} , and ζ_{ji} is the relative transformation between keyframe i and frame j represented by Lie Algebras. At last, with the continuous inputs of the image sequence, LSD-SLAM performs loop closure detection based on the relative pose ζ_{ji} between

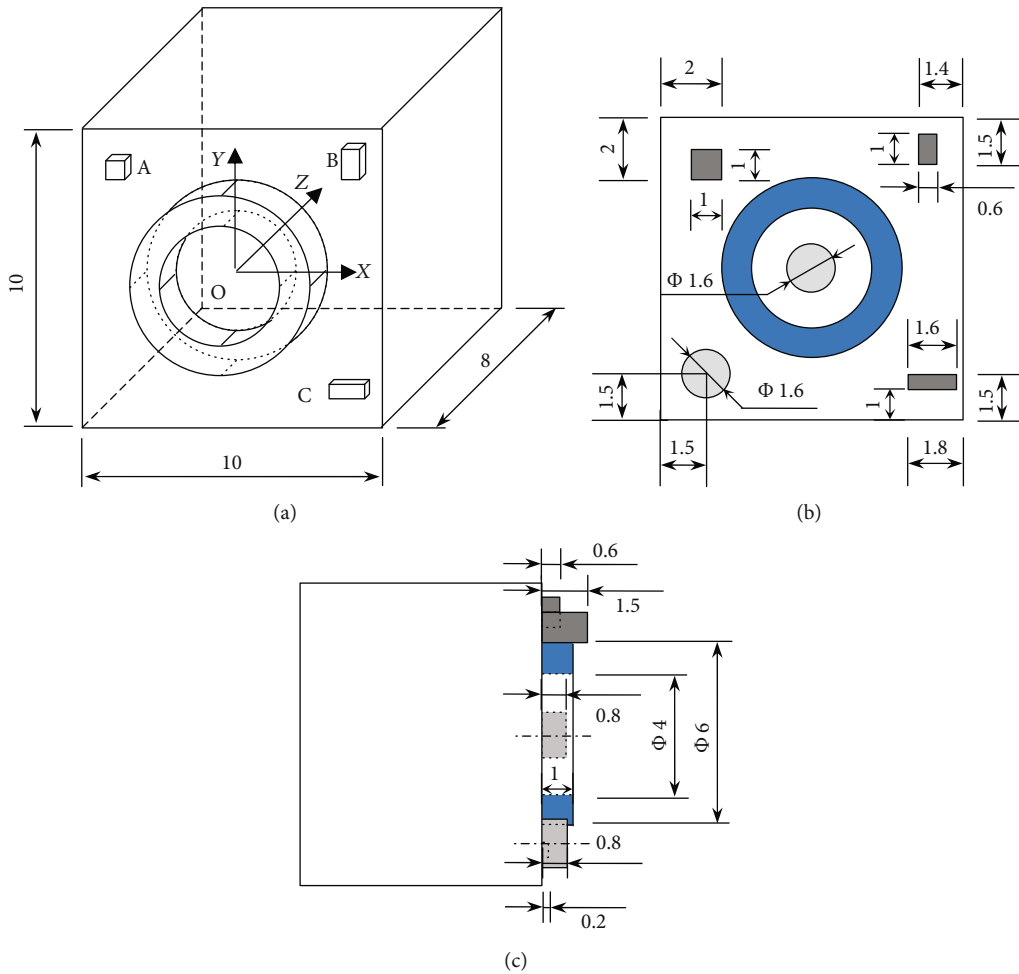


FIGURE 3: Schematic diagram of the satellite model: (a) stereogram, (b) view in the Z direction, and (c) view in the X direction.

the keyframes in the back-end, further eliminating the cumulated error induced by the camera motion [34].

It is worth mentioning that in Numerical Simulations and Experiments of this paper, as we have tried, the loop closure has not been detected and the back-end optimization module of the LSD-SLAM algorithm is not triggered.

4. Numerical Simulations and Experiments

To verify the validity of the method applied in this paper, numerical simulations and experiments are carried out. First, the satellite model of the simulation is a cuboid model with a docking ring mounted on the front and some small satellite components, generated by POV-Ray (<http://www.povray.org/>) software. The model rotates around a single axis and the images of the model are saved at the same time. The calculated results can be obtained by calculating the images by the proposed method. The theoretical results of the rotation angles of the model can be obtained by the software. The two results are compared, and the errors are analyzed.

The satellite model of the experiments is a satellite model fixed on a rack and can rotate around a single axis driven by a stepper motor. The model rotates and the images of the

model are saved at the same time. The calculated results can also be obtained by calculating the images by the proposed method. The theoretical results of the rotation angles of the model can be obtained by the motor. The two results are compared, and the errors are also analyzed.

4.1. Numerical Simulations. In this section, numerical simulations are performed to verify the validity of the proposed method in this paper. As shown in Figure 3, the space coordinate system O - XYZ is established at the center of the front surface of the satellite, and the lengths in the three directions X , Y , and Z of the main satellite body are 10, 10, and 8, respectively (note: this article uses a dimensionless unit). The docking ring mounted on the front surface of the satellite takes O as the center of the circle, and the inner and outer radius, and the height is 2, 3, and 1, respectively. There are three different components on the surface of the satellite named A, B, and C. The lengths in the three directions X , Y , and Z of A are 1, 1, and 1.5, respectively. The lengths in the three directions X , Y , and Z of B are 0.6, 1, and 0.6, respectively. The lengths in the three directions X , Y , and Z of C are 1.6, 0.5, and 0.2, respectively. The camera model in the software is pinhole camera model. Before the simulation,

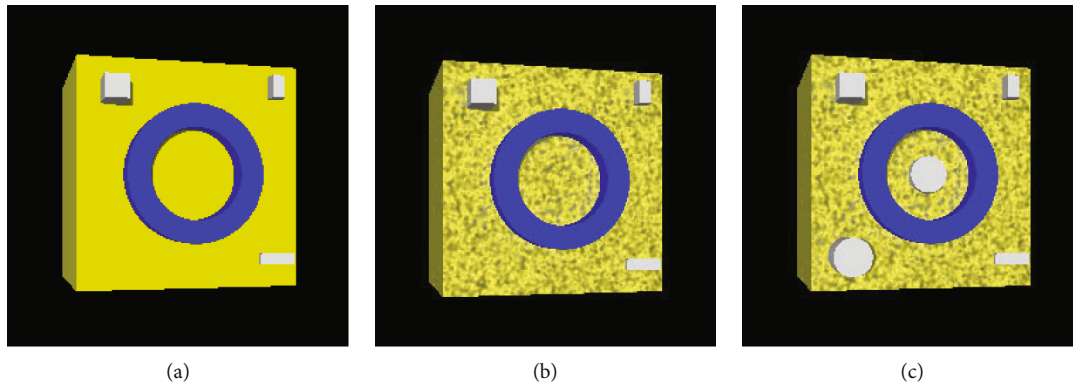


FIGURE 4: Schematic diagram of three different satellite models: (a) no texture with 3 components, (b) rough textures with 3 components, and (c) rough textures with 5 components.

the camera in the software needs to be calibrated. The procedure of calibration is as follows: (1) first, a total of 10-20 images (15 in this paper) of different poses with chessboard are generated and saved as jpg format; (2) then, the corner points of these checkerboard pictures are extracted by using the Camera Calibration Toolbox of MATLAB; (3) at last, the camera parameters are obtained by calculation. The calculated camera parameters are as follows: $f_x = 479$ mm, $f_y = 479$ mm, $c_x = 319$, and $c_y = 239$. The camera parameters do not change in the process of camera movements, so the obtained camera parameters can be used for the subsequent simulations.

In order to indicate the estimation accuracy of kinematic information in different roughness conditions on the surface of the object, this paper considers three kinds of satellite models with different surface texture features and different amounts of the components, as shown in Figure 4, which are (a) no texture on the surface and 3 components mounted on the model, (b) rough textures on the surface and 3 components mounted on the model, and (c) rough textures on the surface and 5 components mounted on the model. There are two additional components in (c), where the radius and height of the cylinder located at the center on the front surface are 0.5 and 1.8, and the radius and height of the cylinder located at the bottom left corner on the front surface are both 0.8. All of the generated images are saved as jpg format, and the size of each picture is 640×480 .

In the first case, it is assumed that the satellite rotates at a constant imaging speed around the Z axis. The imaging speed of the camera is set to be 10 frames per degree. The rotation angles are 90, 180, 360, and 450 degrees, respectively. The POV-Ray could generate sequences of images at different angles under such condition. The camera parameters and sequence image sets are used as input of the LSD-SLAM algorithm, and the estimated angles of the satellite model are calculated and compared with the theoretical values. In Figures 5–10, as the number of frames increases, the theoretical value of the rotation angle increases proportionally, which is expressed in a dashed curve named “theoretical result.” And as the number of frames increases, the calculated value of the rotation angle also increases correspondingly, which

is expressed in a solid curve called “calculated result.” Figures 5–7 show the curves in the first case in the three different models, respectively. And the results of the root mean square (RMS) error, the absolute error, and the fractional error are shown in Table 1. It can be seen from the results that the method presented in this paper can accurately estimate the rotation angle of the satellite.

Then, the second case where the imaging speed of the camera varies should be considered. Assume that the imaging speed is set to be 20, 30, and 40 frames per degree, and the rotation angle of the satellite is 90 degrees. Figures 8–10 show the curves under these conditions in the three different models, respectively. The error results are shown in Table 2. It can be seen from the calculation that the method can still effectively estimate the pose of the satellite.

According to the above results of simulations, the analyses are as follows. (1) The error of angular value obtained by calculating the satellite with rough surface is less than that with no texture. (2) In the condition of 10 frames per degree unchanged, as the rotation angle of the satellite increases, the error of the angular calculation first decreases and then increases; the error is the smallest and reaches the highest precision when the rotation angle is 180 degrees. (3) Under the condition that the satellite rotation is maintained at 90 degrees, as the imaging speed increases, the angular error of the surface with no texture increases, and that of the surface with textures is reduced. In general, the angular calculation error of the satellite model is within a certain range in all cases, which shows that the algorithm has satisfied estimation accuracy.

4.2. Experiments. Here, experiments are performed to verify the validity of the proposed method in this paper. As shown in Figure 11, a satellite model is fixed on a rack and the model can rotate around the axis perpendicular to the plane of the rack driven by a stepper motor. A docking ring and three components are mounted on the front of the satellite model, and two sailboards are on both sides of the satellite. The front surface of the model is wrapped with gold foil. The monocular camera is stationary in front of the satellite model. Since the experiments are for measuring the rotation angle

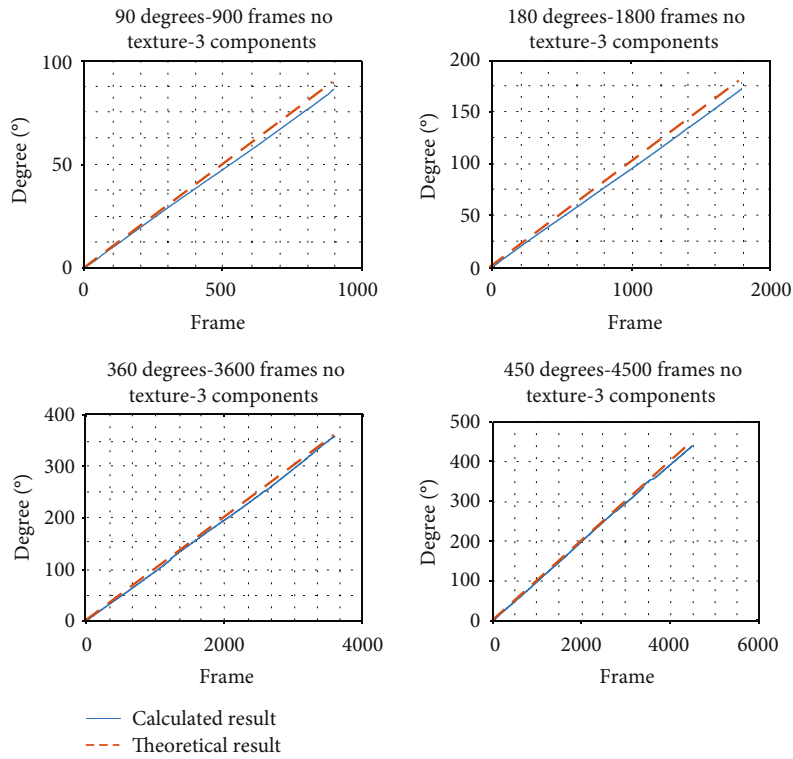


FIGURE 5: Angle estimation of model (a) under different rotation angles (10 frames per degree).

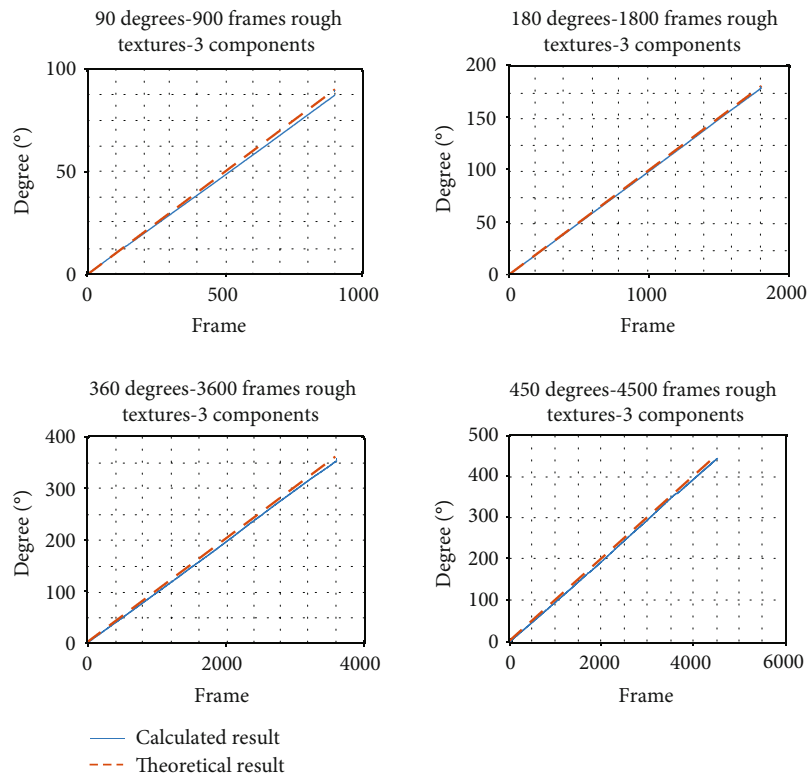


FIGURE 6: Angle estimation of model (b) under different rotation angles (10 frames per degree).

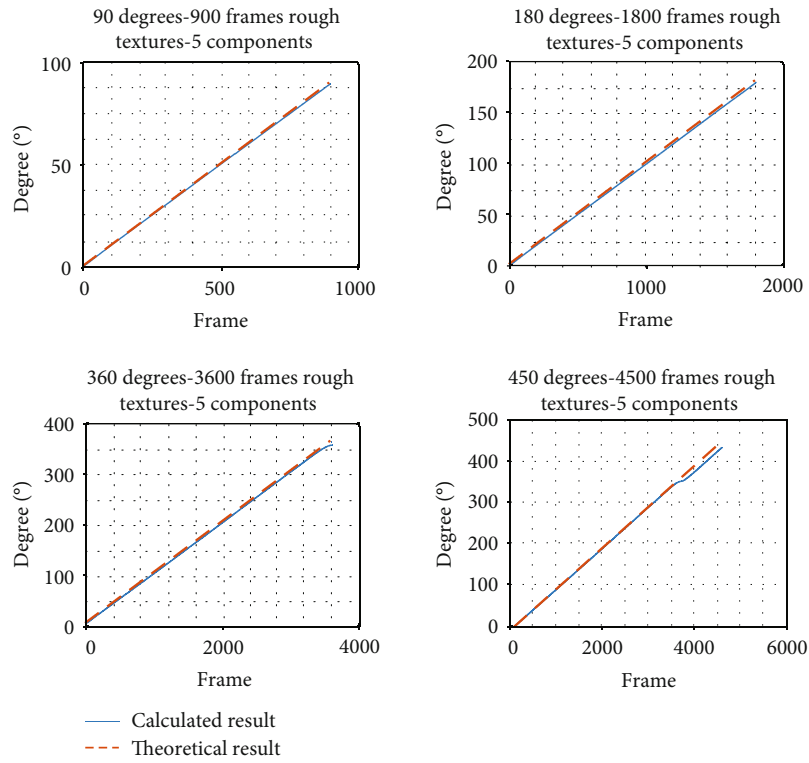


FIGURE 7: Angle estimation of model (c) under different rotation angles (10 frames per degree).

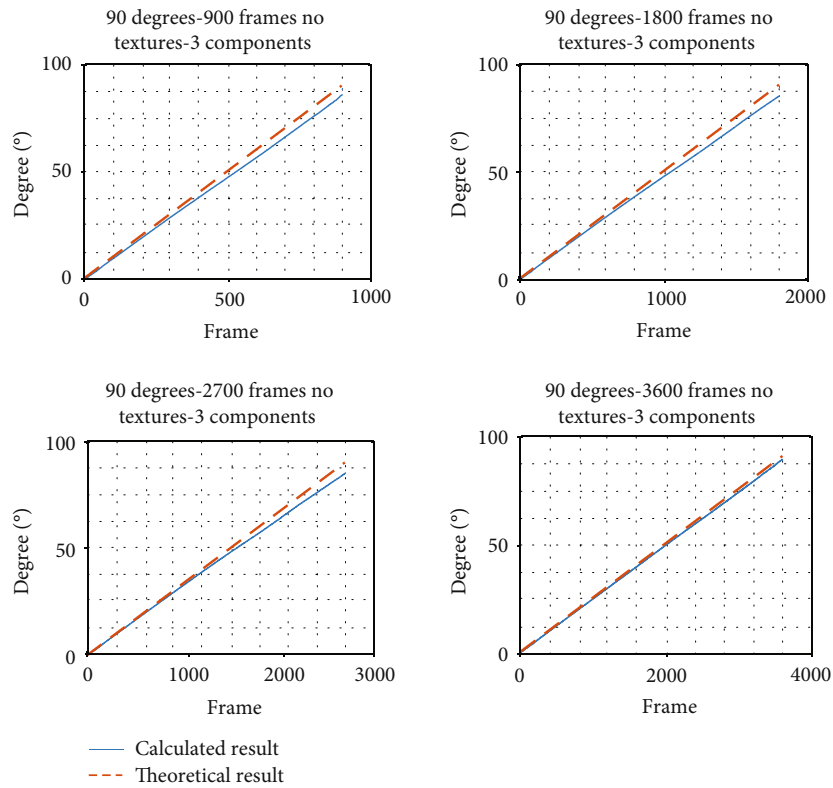


FIGURE 8: Angle estimation of model (a) under different imaging speeds (rotation angle of 90 degrees).

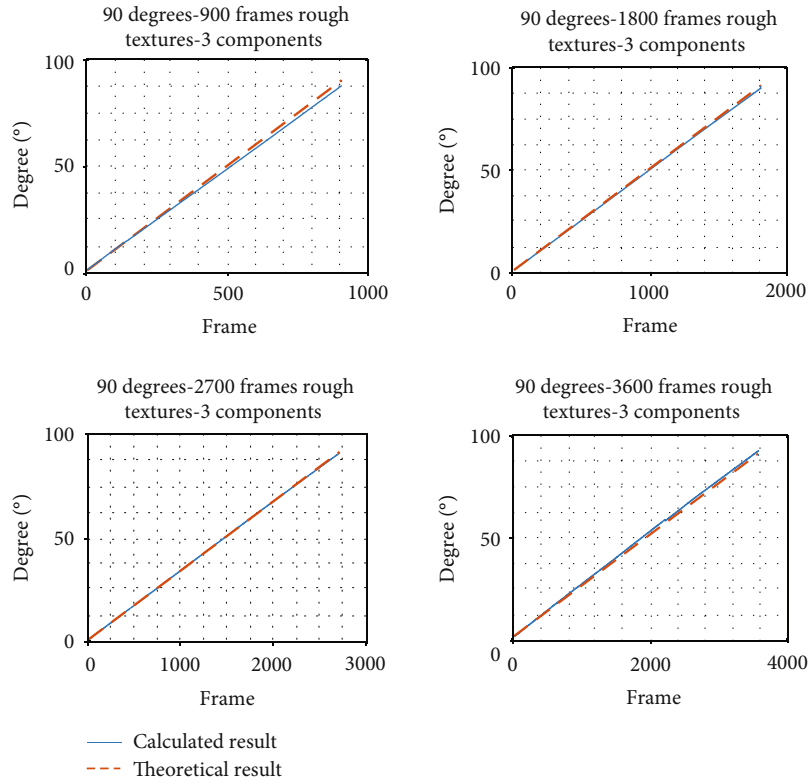


FIGURE 9: Angle estimation of model (b) under different imaging speeds (rotation angle of 90 degrees).

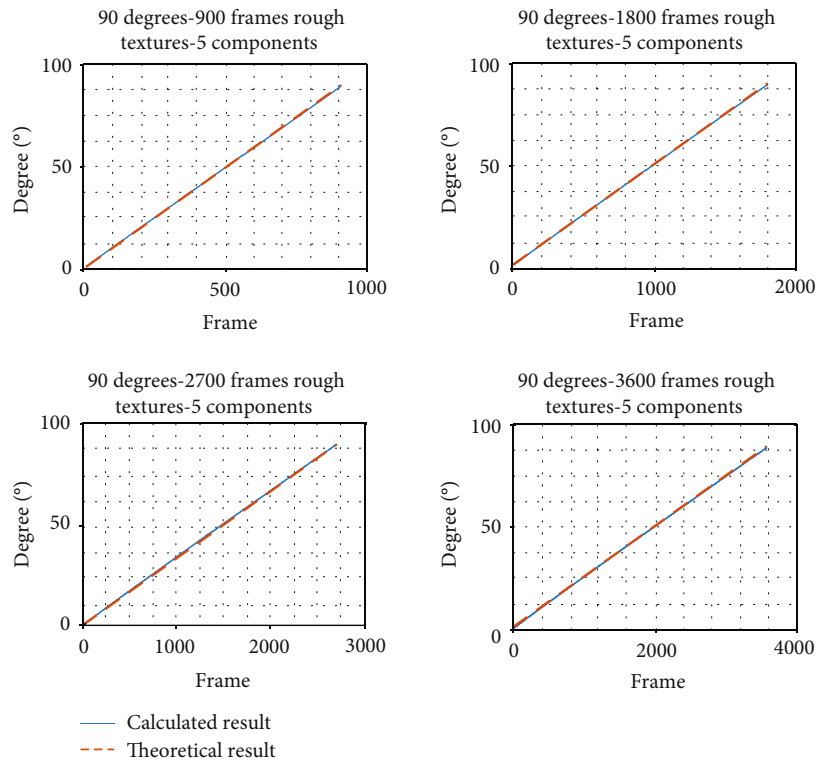


FIGURE 10: Angle estimation of model (c) under different imaging speeds (rotation angle of 90 degrees).

TABLE 1: Calculated values and errors of the three models under different rotation angles (10 frames per degree).

Angle (°)	Calculated value (°)	RMS error (°)	Absolute error (°)	Fractional error (%)
Model (a): no textures_3 components				
90	86.35	2.42	3.65	4.06
180	171.83	5.28	8.17	4.54
360	357.64	4.48	2.36	0.66
450	439.84	4.21	10.16	2.26
Model (b): rough textures_3 components				
90	87.17	1.77	2.83	3.14
180	177.99	1.18	2.01	1.12
360	351.59	4.59	8.41	2.34
450	443.01	4.22	6.99	1.55
Model (c): rough textures_5 components				
90	89.40	0.29	0.60	0.67
180	178.94	0.63	1.06	0.59
360	350.62	1.37	9.38	2.61
450	434.34	6.69	15.66	3.48

TABLE 2: Calculated values and the errors of three different models under different imaging speeds (rotation angle of 90 degrees).

Frame	Calculated value (°)	RMS error (°)	Absolute error (°)	Fractional error (%)
Model (a): no textures_3 components				
900	86.35	2.43	3.65	4.06
1800	85.26	2.46	4.74	5.27
2700	85.05	2.39	4.95	5.50
3600	88.95	0.71	1.05	1.17
Model (b): rough textures_3 components				
900	87.17	1.77	2.83	3.14
1800	88.92	0.49	1.08	1.20
2700	89.42	0.17	0.58	0.64
3600	91.50	1.37	1.50	1.67
Model (c): rough textures_5 components				
900	89.40	0.29	0.60	0.66
1800	89.71	0.23	0.29	0.33
2700	90.06	0.50	0.06	0.06
3600	90.04	0.52	0.04	0.04

perpendicular to the plane of the rack, the specific distance between the model and the monocular camera is no need to know. The relative position between the model and the camera is as shown in Figure 12, where the red frame represents the monocular camera. In order to simulate space environment and reduce interferences, the background is covered by black curtains and the satellite model becomes the only object in the field of view of the camera. When the satellite model rotates driven by the motor, the monocular camera captures the images of the satellite model. The calibrated parameters of the monocular camera are $f_x = 552.38$ mm, $f_y = 554.18$ mm, $c_x = 303.02$, and $c_y = 224.06$. An image captured by the camera is as shown in Figure 13 and the running

process of the algorithm is as shown in Figure 14, where the red frame represents the camera and the green curve represents the trajectory of the camera relative to the model. As the input of the LSD-SLAM algorithm, the images are computed and the experiment results of the rotation angles can be obtained. The theoretical results of the rotation angles can be obtained by the stepper motor. Comparison of the experiment results and the theoretical results is as shown in Figure 15 and Table 3.

According to the above results of experiments, it can be seen that the angular calculation error of the satellite model is within a reasonable range, which shows that the algorithm has satisfied estimation accuracy.

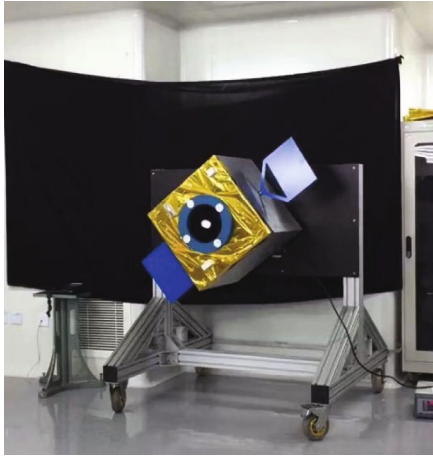


FIGURE 11: Schematic diagram of the satellite model and the experimental scenario.

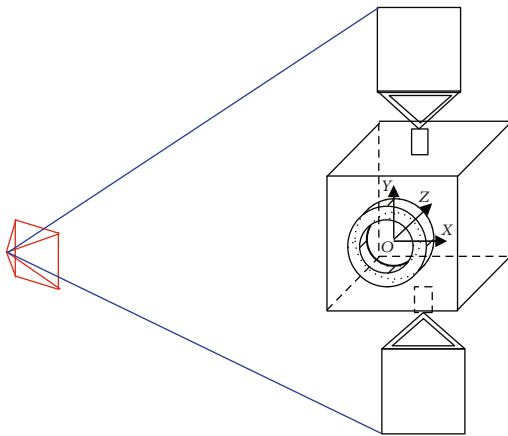


FIGURE 12: Schematic diagram of relative position relationship between the satellite model and the monocular camera.



FIGURE 13: An image of the satellite model captured by the monocular camera.

4.3. *Analysis and Discussion.* According to the result above, the feasibility of the algorithm and the possible causes of errors are analyzed, while other schemes of motion observation are also briefly introduced and discussed.

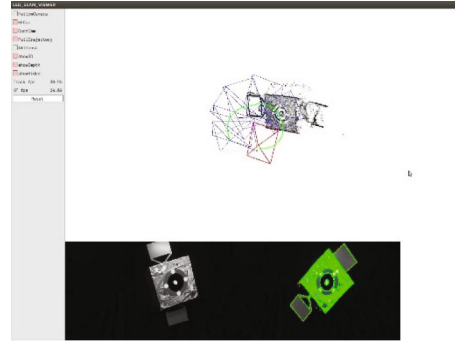


FIGURE 14: Schematic diagram of the running process of the algorithm.

In the above numerical simulations and experiments, the LSD-SLAM algorithm runs on the computer of which the CPU is Inter i7-4790 and the operation system is Ubuntu 16.04. In general, when the frame rate of the camera is 30 frames per second, in order to meet the real-time requirements, the processing time of each frame should not exceed 33.3 ms. Running the algorithm on this computer, the average processing time of each frame is no more than 30 ms, which guarantees the real-time operation of the algorithm.

Based on the above results of numerical simulations and experiments, it can be found that as the frame number and imaging speed increase, the error increases gradually. The possible reason is related to the characteristic of the direct method. Since the corresponding relationship only depends on the intensity, one pixel may have more than one corresponding pixels, which may lead to mismatching. And the photometric residual is based on the hypothesis that the intensity of the same pixel remains unchanged in different images, which is a strong assumption.

There are many schemes for motion observation of non-cooperative targets, such as methods based on radar and lidar. Radar and lidar have the advantages of active measurement, high accuracy, strong direction, and fast observation. The disadvantages are also obvious, such as sensitive to light and radiation, small measurement range, and low efficiency of searching targets. Vision-based scheme has the characteristics of low cost, easy to install, and wide application scenarios, and relatively sensitive to interference and has lower accuracy. Therefore, to have complementary advantages, different sensor fusion becomes the trend of researches on motion observation of noncooperative targets.

5. Conclusion

The vision-based pose estimation of noncooperative space targets is of great scientific and engineering significance for the on-orbit service missions. In order to alleviate the dependence of most existing methods on target features, this paper presents an LSD-SLAM algorithm for pose estimation based on image photometric residuals. The algorithm utilizes the pixels of significant change in gradient to minimize the photometric residuals of pixels to estimate the pose, recover the depth, and construct the semidense three-dimensional map. Considering the error accumulation in long-term

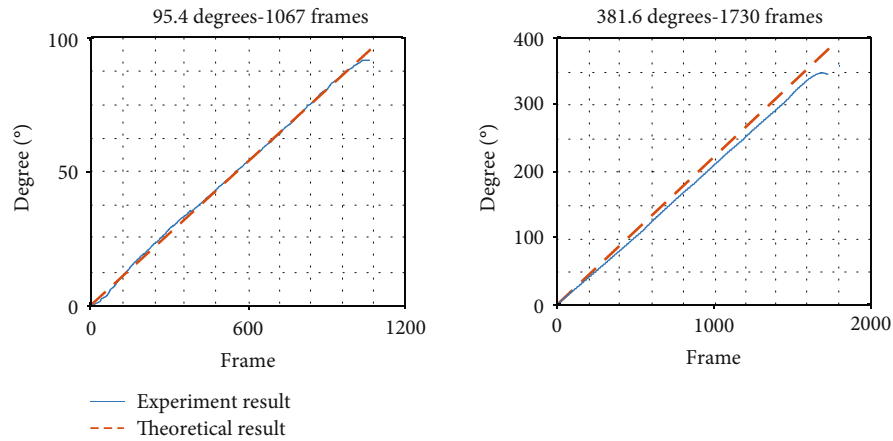


FIGURE 15: Angle estimation results of satellite model in experiments.

TABLE 3: Comparison of theoretical results and calculated results.

Theoretical value (°)	Experiment value (°)	Absolute error (°)	Fractional error (%)
95.4	91.8	3.6	3.8
381.6	360.4	21.2	5.6

observation, the proposed algorithm remains high accuracy in short-term and close range motion observation of a non-cooperative target, especially when there are rough textures on the satellite surface.

However, it should be pointed out that the back-end optimization module of the LSD-SLAM algorithm is not triggered in the simulations and experiments in this paper, which affects the accuracy. The further research will focus on solving this problem and improving the accuracy of calculation.

Data Availability

(1) The images used to verify the effectiveness of the algorithm in the simulations and to support the findings of this study were generated by POV-Ray software (<http://www.povray.org/>). The image data used to support the findings of this study are included within the supplementary information file. Please see the attachment. (2) The curve data and table data used to support the findings of this study were generated by running the LSD-SLAM algorithm using images generated by POV-Ray. LSD-SLAM is an open source program hosted on GitHub, which is proposed and written by Professors Jakob Engel and Daniel Cremers, and the Computer Vision Group of Technical University of Munich. The website of the project is https://github.com/tum-vision/lsd_slam.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

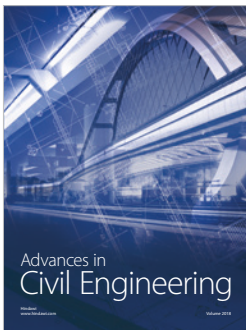
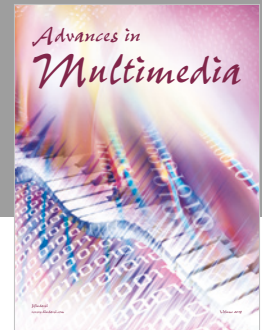
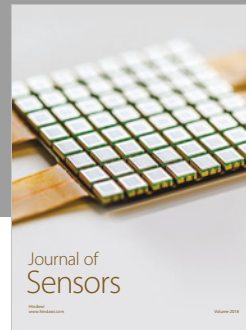
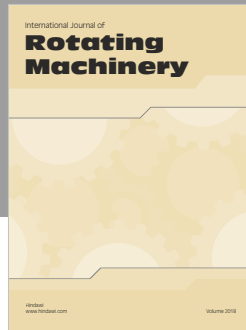
Acknowledgments

This work was supported by the Natural Science Foundation of China (grant numbers 11772187 and 11802174), the China Postdoctoral Science Foundation (grant number 2018M632104), and Shanghai Institute of Technical Physics of the Chinese Academy of Science (grant number CASIR201702).

References

- [1] D. Modenini, M. Zannoni, R. L. Manghi, and P. Tortora, "An analytical approach to autonomous optical navigation for a cubesat mission to a binary asteroid system," *Advances in the Astronautical Sciences*, vol. 163, pp. 139–149, 2018.
- [2] J. A. Christian, "Accurate planetary limb localization for image-based spacecraft navigation," *Journal of Spacecraft and Rockets*, vol. 54, no. 3, pp. 708–730, 2017.
- [3] K. Kanani, A. Petit, E. Marchand, T. Chabot, and B. Gerber, "Vision based navigation for debris removal missions," in *Proceedings of the 63rd International Astronautical Congress*, Naples, Italy, October 2012.
- [4] A. Petit, E. Marchand, and K. Kanani, "Vision-based detection and tracking for space navigation in a rendezvous context," in *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Turin, Italy, September 2012.
- [5] D. Conway, B. Macomber, K. A. Cavalieri, and J. L. Junkins, "Vision-based relative navigation filter for asteroid rendezvous," in *Proceedings of the AAS Guidance, Navigation, and Control Conference*, Breckenridge, CO, February 2014.
- [6] G. Liu, C. Xu, Y. Zhu, and J. Zhao, "Monocular vision-based pose determination in close proximity for low impact docking," *Sensors*, vol. 19, no. 15, p. 3261, 2019.
- [7] S. Sharma, J. Ventura, and S. D'Amico, "Robust model-based monocular pose initialization for noncooperative spacecraft rendezvous," *Journal of Spacecraft and Rockets*, vol. 55, no. 6, pp. 1414–1429, 2018.
- [8] G. Liu, C. Xu, Y. Zhu, and J. Zhao, "Robust, fast and accurate vision-based localization of a cooperative target used for space robotic arm," *Acta Astronautica*, vol. 136, pp. 101–114, 2017.

- [9] J. Z. Song and C. X. Cao, "Pose Self-Measurement of Noncooperative Spacecraft Based on Solar Panel Triangle Structure," *Journal of Robotics*, vol. 2015, Article ID 472461, 6 pages, 2015.
- [10] L. Regoli, K. Ravandoor, M. Schmidt, and K. Schilling, "On-line robust pose estimation for rendezvous and docking in space using photonic mixer devices," *Acta Astronautica*, vol. 96, pp. 159–165, 2014.
- [11] S. D'Amico, "Pose estimation of an uncooperative spacecraft from actual space imagery," *International Journal of Space Science and Engineering*, vol. 2, no. 2, pp. 171–189, 2014.
- [12] R. Li, Y. Zhou, F. Chen, and Y. Chen, "Parallel vision-based pose estimation for non-cooperative spacecraft," *Advances in Mechanical Engineering*, vol. 7, no. 7, 2015.
- [13] T. Shtark and P. Gurfil, "Tracking a non-cooperative target using real-time Stereovision-Based control: an experimental study," *Sensors*, vol. 17, no. 4, p. 735, 2017.
- [14] Y. He, B. Liang, X. Du, X. Wang, and D. Zhang, "Measurement of relative pose between two non-cooperative spacecrafts based on graph cut theory," in *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, pp. 1900–1905, IEEE, 2014.
- [15] G. Dong and Z. H. Zhu, "Vision-based pose and motion estimation of non-cooperative target for space robotic manipulators," in *AIAA Space Conference and Exposition*, San Diego, USA, 2014.
- [16] D. Mortari, C. N. D'Souza, and R. Zanetti, "Image processing of illuminated ellipsoid," *Journal of Spacecraft and Rockets*, vol. 53, no. 3, pp. 448–456, 2016.
- [17] D. Modenini, "Attitude determination from ellipsoid observations: a modified orthogonal Procrustes problem," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 10, pp. 2324–2326, 2018.
- [18] C. Liu and W. Hu, "Relative pose estimation for cylinder-shaped spacecrafts using single image," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 4, pp. 3036–3056, 2014.
- [19] H. Zhang, Z. Jiang, Y. Yao, and G. Meng, "Vision-based pose estimation for space objects by Gaussian process regression," in *2015 IEEE Aerospace Conference*, pp. 1–9, IEEE, 2015.
- [20] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.
- [21] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.
- [22] B. E. Tweddle, "Computer vision-based localization and mapping of an unknown, uncooperative and spinning target for spacecraft proximity operations," *PhD thesis*, Massachusetts Institute of Technology, Cambridge, MA, USA, 2013.
- [23] S. Augenstein and S. M. Rock, "Improved frame-to-frame pose tracking during vision only SLAM/SFM with a tumbling target," in *proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3131–3138, Shanghai, China, May 2011.
- [24] S. Chiodini, R. G. Reid, B. Hockman, I. A. D. Nesnas, S. Debei, and M. Pavone, "Robust visual localization for hopping rovers on small bodies," in *IEEE International Conference on Robotics and Automation*, pp. 897–903, Brisbane, Australia, May 2018.
- [25] M. Dor and P. Tsiotras, "ORB-SLAM applied to spacecraft non-cooperative rendezvous," in *Space Flight Mechanics Meeting*, Kissimmee, USA, 2018.
- [26] X. D. Du, Y. He, L. Chen, and S. Gao, "Pose estimation of large non-cooperative spacecraft based on extended PnP model," in *IEEE International Conference on Robotics and Biomimetics*, Qingdao, China, 2016.
- [27] H. M. Liu, G. F. Zhang, and H. J. Bao, "A review of simultaneous localization and mapping method based on monocular vision," *Journal of Computer Aided Design & Computer Graphics*, vol. 28, no. 6, pp. 855–866, 2016.
- [28] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision (Second Edition)*, Cambridge University Press, Cambridge, 2003.
- [29] R. A. Newcome, S. J. Lovegrove, and A. J. Davison, "DTAM: dense tracking and mapping in real-time," in *IEEE International Conference on Computer Vision*, pp. 2320–2327, Barcelona, Spain, 2011.
- [30] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: large-scale direct monocular slam," in *European Conference on Computer Vision*, pp. 834–839, Zurich, Switzerland, 2014.
- [31] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," in *Proceedings of the IEEE international conference on computer vision*, pp. 1449–1456, Sydney, Australia, December 2013.
- [32] T. D. Barfoot, *State Estimation for Robotics: Matrix Lie Groups*, Cambridge University Press, Cambridge, 2017.
- [33] G. Q. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Analysis and improvement of the consistency of extended Kalman filter based slam," in *IEEE International Conference on Robotics and Automation*, pp. 473–479, Pasadena, USA, 2008.
- [34] A. Glover, W. Maddern, M. Warren, S. Reid, M. Milford, and G. Wyeth, "OpenFABMAP: an open source toolbox for appearance-based loop closure detection," in *IEEE International Conference on Robotics and Automation*, pp. 4730–4735, Minnesota, USA, 2012.



Hindawi

Submit your manuscripts at
www.hindawi.com

