World Academy of Science, Engineering and Technology
International Journal of Cognitive and Language Sciences
Vol:17, No:12, 2023

Acceleration-Based Motion Model for Visual SLAM

Daohong Yang, Xiang Zhang, Wanting Zhou, Lei Li

Abstract—Visual Simultaneous Localization and Mapping (VSLAM) is a technology that gathers information about the surrounding environment to ascertain its own position and create a map. It is widely used in computer vision, robotics, and various other fields. Many visual SLAM systems, such as OBSLAM3, utilize a constant velocity motion model. The utilization of this model facilitates the determination of the initial pose of the current frame, thereby enhancing the efficiency and precision of feature matching. However, it is often difficult to satisfy the constant velocity motion model in actual situations. This can result in a significant deviation between the obtained initial pose and the true value, leading to errors in nonlinear optimization results. Therefore, this paper proposes a motion model based on acceleration that can be applied to most SLAM systems. To provide a more accurate description of the camera pose acceleration, we separate the pose transformation matrix into its rotation matrix and translation vector components. The rotation matrix is now represented by a rotation vector. We assume that, over a short period, the changes in rotating angular velocity and translation vector remain constant. Based on this assumption, the initial pose of the current frame is estimated. In addition, the error of the constant velocity model is analyzed theoretically. Finally, we apply our proposed approach to the ORBSLAM3 system and evaluate two sets of sequences from the TUM datasets. The results show that our proposed method has a more accurate initial pose estimation, resulting in an improvement of 6.61% and 6.46% in the accuracy of the ORBSLAM3 system on the two test sequences, respectively.

Keywords—Error estimation, constant acceleration motion model, pose estimation, visual SLAM.

I. Introduction

IN the fields of computer vision, robotics, and autonomous driving, Simultaneous Localization and Mapping (SLAM) plays a crucial role. The underlying principle is that the utilization of a robot as a representation allows for the integration of a sensor within the robot, enabling it to gather environmental data. This, in turn, empowers the robot to perform tasks such as positioning and mapping. VSLAM is a technique that utilizes a camera as a sensor, which is a commonly employed sensor capable of capturing a vast amount of image data. At present, there are many mature types of research on VSLAM, and its basic framework mainly includes front end, back end, mapping, and loop detection [1]. The front end is also known as the visual odometer, whose main function is to obtain preliminary self-pose information and environmental map information by analyzing and calculating the input image data. The back end receives camera poses from visual odometer measurements at different times and optimizes them to get globally consistent trajectories and maps. Mapping involves the creation of an environment map. The purpose of loop detection is to mitigate the impact of accumulated errors. VSLAM can be classified into two primary methods: the feature point method and the direct method, which differ in their employed approaches. Among the various technologies, VSLAM based on the feature point method involves the extraction of key points from images and establishing matching relationships between points from different frames. This enables the calculation of the camera's motion position. This particular approach has undergone thorough investigation, with ORBSLAM3 [2] being recognized as one of the most efficient features of point-based visual SLAM systems currently available. The main components of this system consist of three interconnected threads, namely image tracking, local mapping, and closed-loop detection. The image tracking part is the main thread of the system. Due to the intricate nature of extracting and matching feature points, ORBSLAM3 utilizes a constant velocity motion model [3], a widely adopted approach in numerous visual SLAM systems, to enhance the efficiency of tracking and matching between successive frames. The constant velocity motion model makes an estimation of the initial pose value of the current frame by assuming that the pose undergoes the same change between adjacent frames. This assumption can be met when the frame rate of the image sequence is high or when the motion speed is slow. The complexity of the camera's actual motion state can be significant. For instance, the camera has the potential to undergo rapid acceleration or deceleration within a brief time frame. In this particular scenario, meeting the assumptions of the constant speed motion model proves challenging, thereby frequently resulting in the failure to accurately track the object.

Therefore, we propose a model of motion with constant acceleration to address this issue. When the camera undergoes rapid acceleration or deceleration, it can be assumed that the acceleration of the camera movement remains constant over a short period of time. In such cases, the initial pose can be determined by analyzing the acceleration. The primary contributions of this paper are outlined as follows.

- We provide a theoretical analysis of the pose estimation error for the constant velocity motion model.
- We propose a constant acceleration motion model that can be effectively employed in most SLAM systems. We make the assumption that the acceleration between adjacent frames remains constant within a brief time period. The pose transformation matrix is decoupled, and the initial values of the rotation matrix and translation vector are estimated separately. The computational complexity can be reduced by converting the estimation of the rotation data matrix to that of the rotation vector.

Daohong Yang, Wanting Zhou and Lei Li are with the Electronic Information, University of Electronic Science and Technology of China, Chengdu, Sichuan, China (e-mail: yangdhll@163.com, zhouwt@uestc.edu.cn).

Xiang Zhang is with the Integrated Circuit Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan, China.

- We employ the constant acceleration motion model to analyze the performance of ORBSLAM3 on TUM datasets. The obtained outcome exhibited superior performance compared to the constant speed model of ORBSLAM3.

II. Related Work

Visual SLAM has a rich historical background and has evolved into a sophisticated and well-established system. Davison et al. introduced MonoSLAM [4], the first monocular VSLAM approach. They adopted monocular camera as the sole source of data for their system and proposed a 3D map that relied on probabilistic features. The state variable consisted of the camera's attitude and the three-dimensional coordinates of the map points. Assuming that the state vector followed a multi-dimensional Gaussian distribution, the map data were updated with the Extended Kalman Filter (EKF) [5] as the camera's attitude changed. There algorithm lacks the capability to perform large-scale map construction and there is no loop detection, back-end optimization, and other modules in the system.

Klein et al. introduced PTAM (Parallel Tracking and Mapping) [6], which represents the pioneering VSLAM system. Notably, PTAM is distinguished by its ability to autonomously execute the tracking and mapping processes concurrently, utilizing two parallel threads. Besides, they incorporated the concept of key frame into the algorithm and employed it to depict the camera pose in the map. FAST corner points [7] were employed to establish data association between two consecutive key frames based on the analysis of reprojection errors. The author incorporated the BA (Bundle Adjustment) optimization method [8] in updating the local map and global map. Subsequent research largely adhered to this framework.

Klein et al. introduced ORB-SLAM [9], a monocular real-time SLAM system that utilizes ORB feature matching. This system represents a significant advancement in the field of sparse VSLAM. The system is very perfect and can be applied to a variety of scenarios. The proposed method exhibits a high level of robustness when dealing with moving clutter. Additionally, it has the functions of tracking, mapping, relocation and loop detection. It uses three threads symbolically, which are the feature point tracking thread, the local reprojection error optimization thread, and the global optimization thread based on the pose map. The process of selecting rebuild points and key frames was found to be robust, enabling the generation of incremental maps. This advancement played a significant role in establishing feature point-based SLAM as the prevailing approach during that period.

ORB-SLAM2 [10], which is a derivative of ORB-SLAM, has been developed to support real-time camera tracking using monocular, stereo, and RGB-D cameras on the CPU. ORB-SLAM2 incorporates a comprehensive approach by integrating monocular, stereoscopic, and RGB-D methods. It also achieves global optimization and closed-loop detection technology. However, in the event that the system fails to identify frames with high similarity, it can lead to tracking failures and subsequent loss of state.

The Dense Tracking and Mapping (DTAM) system [11] described herein represents the pioneering implementation of the direct method and is capable of building dense maps for every frame of images. The algorithm yields a precise and comprehensive reconstruction, however, the level of density reconstruction significantly impacts the computational expenses associated with data storage and processing. Therefore, in order to achieve real-time operation, the algorithm needs to be on the GPU to realize real-time calculation. Then, Kinect Fusion system [12] based on RGB-D camera is proposed. The truncated signed distance function (TSDF) algorithm is used to represent pixel grid, and iterative closest point (ICP) is adopted. The algorithm is the first to operate in real time based on RGB-D sensors. The algorithm consists of four primary steps, namely measurement, pose estimation, reconstruction update, and surface prediction. The system exhibits a defect of cumulative drift error due to the absence of loop detection. The Semi-Direct Visual Odometry (SVO) [13] algorithm proposed by Forster et al. combines the advantages of feature-based and direct methods. The algorithm is partitioned into two threads: motion estimation and graph building. The approach employed in this study involves the utilization of direct techniques for tracking and triangulating pixels with a significant image gradient. However, it also relies on joint optimization that is based on feature methods. The semi-direct VO and robust probability depth estimation algorithm can effectively track the corners and edges of pixels. The algorithm can be easily extended to encompass multiple camera tracking, incorporating motion priors. Furthermore, it can be effectively utilized with cameras that possess a wide field of view, such as fisheye and inverse refraction cameras. The advantages of SVO over other VSLAMs are speed and low computational requirements. But without back-end optimization and loopback detection, it is not a complete SLAM system. In 2018, Loo et al. proposed to use neural network to predict the SVO version of monocular image depth [14], which could predict the prediction results of network depth according to the depth of single image, and improve SVO mapping by initializing the mean and variance of depth at feature points.

In 2020, Campos et al. proposed ORB-SLAM3 [2], which was based on ORBSLAM2 [10]. The previous VSLAM schemes were all tightly coupled with the camera model. In this paper, they decoupled the camera model from the VSLAM system, greatly improving the system's suitability to other camera models. This study presents the fisheye lens model and pinhole camera model. Another contribution of the authors is the addition of an inertial sensor (IMUs) to a visual inertial tightly coupled system that relies on a maximum a posteriori probability estimate, which greatly improves the robustness of the system in case of short-term tracking failure. The system implementation also introduced the concept of atlas [15] to realize short, medium and long term data association. ORB-SLAM3 is regarded as one of the most proficient visual SLAM systems currently accessible.

VSLAM systems represented by ORB-SLAM3 [2] adopt a

constant velocity motion model for tracking image frames. This model assumes that the pose change between two adjacent frames remains constant, and calculates the initial pose of the current frame based on this assumption. There are also SVO [13] and LSD-SLAM [16] that directly make the pose of the camera at the current moment equal to that of the previous moment. However, achieving this requirement is often challenging in real-world scenarios. To address this issue, the present paper introduces a motion model.

III. Proposed Method

This section provides a comprehensive explanation of our proposed method. Firstly, this paper presents the tracking thread process of the ORBSLAM3 system. Subsequently, an analysis of the motion model is conducted to determine the initial value error of the pose. Finally, a detailed explanation of our proposed motion model is provided.

A. Tracking Process

The tracking thread mainly includes the preprocessing of input image data, map initialization, tracking module, local map tracking, and key frame determination. The preprocessing part of data is mainly the extraction of ORB feature points. The map initialization module determines the world coordinate system and the initial map points. The tracking module encompasses various tracking techniques, such as motion model tracking, reference key frame tracking, and relocation tracking. The local map tracking module constructs local key frames and optimizes the pose again by using local map points observed in local key frames. Then we select the key frame to enter the local drawing thread. In most cases, the tracking module uses the constant speed motion model to track ordinary frames. When the motion model fails, the tracking is carried out through the reference key frame. If neither of these two ways is successful, the map can only be relocated. Given the real-time constraints of the visual SLAM system, it is imperative to develop an appropriate motion model that can enhance the efficiency of feature point matching.

If a motion model is adopted for tracking, we first need to obtain the initial pose of the current frame according to the motion model, and then use the initial pose to project the map point of the previous frame into the pixel coordinate system of the current frame. Descriptors are then matched within the specified search radius of the projection point. So we have a set of 3D-2D matching points. The matching relationship is utilized to establish BA optimization for solving the pose of the current frame.

B. Constant Acceleration Motion Model

Given that the camera's movement in a real-world environment often involves variable speeds, relying on a constant speed model to estimate the initial pose of the current frame can result in tracking failure. However, it is possible to make an additional assumption, namely, that the rate of change in camera motion, referred to as "acceleration," will remain relatively stable over a brief period of time. In order to enhance the representation of acceleration in this context, we implemented a decoupling approach for estimating the pose transformation matrix. Specifically, we separately estimated the rotation matrix and displacement vector. Therefore, our method primarily relies on two assumptions.

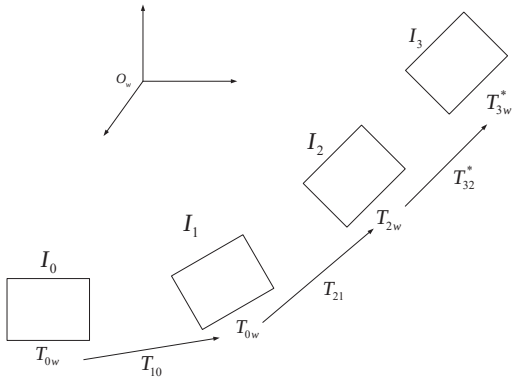


Fig. 1 Constant acceleration motion model

- During the process of estimating the rotation matrix, it is observed that the rotation axis and angular acceleration of camera rotation remain constant over a brief time interval.
- The translation vector remains constant over a brief temporal interval.

As shown in Fig. 1, $I_i, (i = 0, 1, 2, 3)$ respectively represents the i frame image, corresponding to the time of τ_i, where τ_3 is the time of the current frame, that is, I_3 is the current frame, and the corresponding pose of each frame is T_{iw}. It represents the transformation matrix from the world coordinate system to the current frame coordinate system. T^*_{3w} is the pose of the current frame to be calculated. The matrix $T_{ij}(i, j = 0, 1, 2, 3)$ denotes the pose transformation matrix that maps coordinates from frame j to frame i. R_{ij} represents the rotation matrix from frame j to frame i, and $\mathbf{t}_{ij}$ represents the translation vector from frame j to frame i. For the transformation matrix, the following equation holds:

$$T_{iw} = \begin{bmatrix} R_{iw} & \mathbf{t}_{iw} \\ \mathbf{0}^\intercal & 1 \end{bmatrix} \tag{1}$$

In (1), R_{iw} represents the rotation matrix from the world coordinate system to the coordinate system where frame i is located, and $\mathbf{t}_{iw}$ represents the translation vector from the world coordinate system to the current frame coordinate system.

For a rotation described by the rotation matrix R, it is possible to represent it using the rotation vector $\theta \boldsymbol{n}$, where θ represents the rotation angle and $\mathbf{n}$ represents the rotation axis. The relationship between the rotation matrix and the rotation vector can be derived using Rodrigues's Formula as follows:

$$R = I + (1 - \cos\theta)(\mathbf{n}^\wedge)^2 + \sin\theta \mathbf{n}^\wedge \tag{2}$$

Wwhere, $\mathbf{n}^\wedge$ represents the antisymmetric matrix of the vector $\mathbf{n}$.

Therefore, the utilization of a rotation vector is preferred over a rotation matrix. At this juncture, we possess the pose transformation from frame I_j to frame I_i, along with the associated rotation angle θ_{ij}, rotation axis $\mathbf{n}_{ij}$, and translation vector $\mathbf{t}_{ij}$. It is worth mentioning that the following formulas about vector addition and subtraction are carried out in the same coordinate system.

$$\delta\theta = \theta_{21} - \theta_{10} \tag{3}$$

$$\delta\mathbf{t} = \mathbf{t}_{21} - \mathbf{t}_{10} \tag{4}$$

According to the assumptions of the constant acceleration model, let T_{21} to T_{32}^* represent the change in rotation angle $\delta\theta^*$, and let $\delta\mathbf{t}^*$ represent the change in the translation vector.

$$\delta\mathbf{t}^* = \delta\mathbf{t}, \delta\theta^* = \delta\theta \tag{5}$$

Therefore, the rotation vector and translation vector corresponding to T_{32}^* can be expressed as:

$$\theta_{32} = \theta_{21} + \delta\theta^* \tag{6}$$

$$\mathbf{t}_{32}^* = \mathbf{t}_{21} + \delta\mathbf{t}^* \tag{7}$$

According to the assumption made in the constant acceleration model that the rotation axis remains constant over a short period of time, the following can be observed:

$$\mathbf{n}_{32}^* = \mathbf{n}_{21} \tag{8}$$

Finally, the transform matrix can be recovered from the estimation of the rotation vector and the translation vector, which is:

$$T_{32}^* = \begin{bmatrix} R_{32}^* & \mathbf{t}_{32}^* \\ \mathbf{0}^\intercal & 1 \end{bmatrix} \tag{9}$$

where R_{32}^* is calculated by Rodriguez formula (2).

C. Error Estimation

1) Reprojection Search and Matching: In tracking by motion model, it mainly uses the initial pose of the current frame obtained by the motion model for reprojection search and matching. As shown in Fig. 2, the 3D map points that correspond to the feature points of the previous frame are projected onto the current frame. The descriptors are then calculated in the vicinity of the projection points to establish a 3D-2D matching relationship. This relationship is utilized to solve the pose of the current frame through BA optimization. Now we suppose in continuous three moment τ_1, τ_2, τ_3, the corresponding image frame is I_1, I_2, I_3, the corresponding position is T_1, T_2, T_3. By assumptions in the first I_2 frame with image feature points in the p^i, its corresponding map point is p_w^i, the reprojection of map point to the I_3 frame of image point is $\hat{p}^i$, $\hat{p}^i = \pi(T_0 P_w^i)$, where $\pi(\cdot)$ is the projection equation of the camera, and the actual observation point of this map point in frame I_3 is p^j. Now we suppose that I_1 frame to I_2 frame transformation matrix is T_{21}, I_2 to I_3 frame transformation matrix is T_{32}, so the formation type:

$$\begin{aligned} T_{2w} &= T_{21} T_{1w} \\ T_{3w} &= T_{32} T_{2w} \end{aligned} \tag{10}$$

When the camera moves at a constant velocity, the equality $T_{32} = T_{21}$ holds. The Euclidean distance between the actual observation point p^j and the reprojection point $\hat{p}^i$ is typically constrained within a specific range. This range can be represented as a circle with a radius r, such that $\|p^j - \hat{p}^i\|_2 \leqslant r$. However, when the camera moves in variable speed, the Euclidean distance between the actual observation point and the reprojection point may be greater than r, so the algorithm cannot search for the best actual observation point around the reprojection point, which will lead to mismatching.

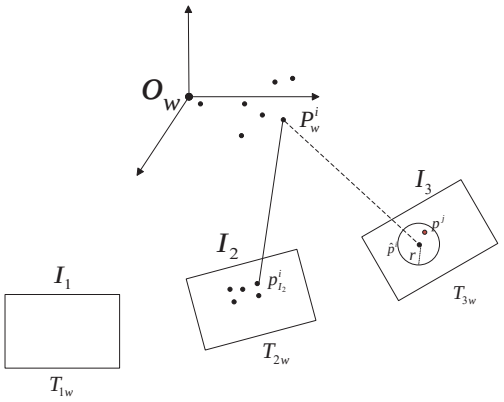


Fig. 2 Reprojection search and matching

2) Error estimation: In this section, the theoretical analysis of the tracking motion model error will be conducted. According to the pinhole model of the camera, the coordinates of the actual observation point p^j and the reprojection point $\hat{p}^i$ can be derived using the following procedure:

$$\begin{aligned} \hat{p}^i &= s^{-1} K T_{21} T_{2w} P_w^i \\ p^j &= s^{-1} K T_{32} T_{2w} P_w^i \end{aligned} \tag{11}$$

where, s represents the depth value of the points, and K is the internal parameter of the camera.

$$\|p^j - \hat{p}^i\|_2 \leqslant \|s^{-1} K T_{32} T_{2w} P_w^i - s^{-1} K T_{21} T_{2w} P_w^i\|_2 \tag{12}$$

Since the 2-norm of the matrix is compatible with the 2-norm of the vector, (12) can be expressed as:

$$\|p^j - \hat{p}^i\|_2 \leqslant s^{-1} \|K\|_{m_2} \cdot \|T_{32} - T_{21}\|_{m_2} \cdot \|T_{2w} P_w^i\|_2 \tag{13}$$

let $c = s^{-1}\|K\|_{m_2}\|T_{2w}P_w^i\|_2$, c is constant.

$$\|p^j - \hat{p}^i\|_2 \leqslant c \cdot \|T_{32} - T_{21}\|_{m_2} \tag{14}$$

Then,

$$\begin{aligned} \|T_{32} - T_{21}\|_{m_2} =& tr((T_{32} - T_{21})^\intercal (T_{32} - T_{21})) \\ =& tr(T_{32}^\intercal T_{32}) - tr(T_{32}^\intercal T_{21}) \\ & - tr(T_{21}^\intercal T_{32}) + tr(T_{21}^\intercal T_{21}) \end{aligned} \tag{15}$$

where $tr(\cdot)$ represents the trace of matrix.

From (9), we can obtain the rotation matrix R_{32} and the translation vector $\mathbf{t}_{32}$ corresponding to T_{32}, R_{21} and $\mathbf{t}_{21}$

corresponding to T_{21}. In addition, since the rotation matrix is orthogonal, $tr(R_{32})^\intercal R_{32} = tr(R_{21})^\intercal R_{21} = 3$. Therefore, let $\boldsymbol{a} = \boldsymbol{t}_{32} - \boldsymbol{t}_{21}$, (15) can be expressed as:

$$\begin{aligned}\|T_{32} - T_{21}\|_{m_2} =& tr(R_{32}^\intercal R_{32}) + \mathbf{t}_{32}^\intercal \mathbf{t}_{32} - 2tr(R_{32}^\intercal R_{21}) \\ & - 2\mathbf{t}_{32}^\intercal \mathbf{t}_{21} + tr(R_{21}^\intercal R_{21}) + \mathbf{t}_{21}^\intercal \mathbf{t}_{21} \\ =& 6 + \mathbf{a}^\intercal \mathbf{a} - 2tr(R_{32}^\intercal R_{21})\end{aligned} \tag{16}$$

Equation (2) shows that R_{32} can be represented by a rotation vector $\theta_{32}\mathbf{n}_{32}$, R_{21} can be represented by a rotation vector $\theta_{21}\mathbf{n}_{21}$. Therefore,

$$\begin{aligned}R_{32}^\intercal R_{21} = (\cos\theta_{32}\mathbf{I} + (1-\cos\theta_{32})\mathbf{n}_{32}\mathbf{n}_{32}^\intercal + \sin\theta_{32}\mathbf{n}_{32}^\wedge)\cdot \\ (\cos\theta_{21}\mathbf{I} + (1-\cos\theta_{21})\mathbf{n}_{21}\mathbf{n}_{21}^\intercal + \sin\theta_{21}\mathbf{n}_{21}^\wedge)\end{aligned} \tag{17}$$

To simplify (17), we get:

$$tr(R_{32}^\intercal R_{21}) = 1 + 2\cos(\theta_{32} - \theta_{21}) \tag{18}$$

Combine (18) and (16), we have:

$$\begin{aligned}\|T_{01} - T_{12}\|_{m_2} &= 6 + \mathbf{a}^\intercal \mathbf{a} - 2(1 + 2\cos(\theta_{32} - \theta_{21})) \\ &= 4 - 4\cos(\theta_{32} - \theta_{21}) + \mathbf{a}^\intercal \mathbf{a}\end{aligned} \tag{19}$$

Finally, we can obtain the Euclidean distance between the actual observation point and the reprojection point in the image coordinate system:

$$\|p^j - \hat{p}^i\|_2 \leqslant c(4 - 4\cos\delta\theta + \mathbf{a}^\intercal \mathbf{a}) \tag{20}$$

where $\delta\theta = \theta_{32} - \theta_{21}$.

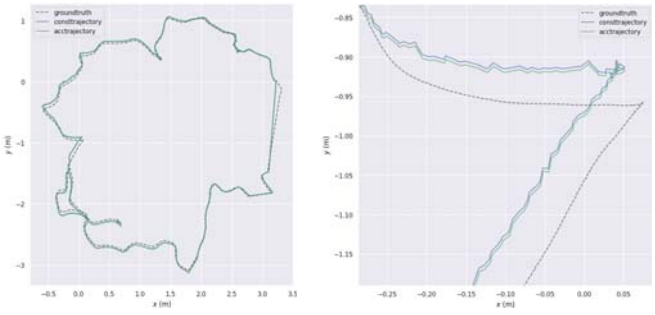
It is not difficult to see from (20) that the accuracy of the transformation matrix from the previous frame to the current frame affects the distance between the actual point and the projected point. Therefore, if we regard the camera motion process as uniform motion, the accuracy of tracking will be affected. And we know that uniform motion can also be viewed as accelerating motion with zero acceleration. In this section, the theoretical analysis of the error associated with tracking using a constant velocity model is presented. From the results, it can be seen that the error is related to the change of rotation angle and the change of translation vector.

IV. Experimental Results

In this section, we have implemented our proposed constant acceleration model to the ORBSLAM3 system and conducted an evaluation of our approach using the TUM datasets. The proposed method was compared with the ORBSLAM3 system using the constant speed model. In addition, the evo evaluation tool [17] was used to calculate the absolute and relative trajectory errors. The findings indicate that the accuracy of the ORBSLAM3 system is enhanced by our proposed method. All experiments were performed on a PC with an Intel i5-1035G1 CPU and 16GB RAM.

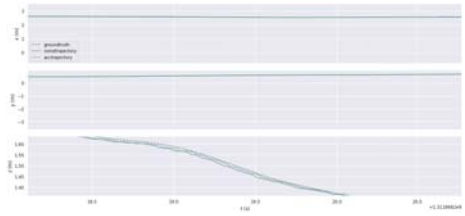
A. TUM Datasets

The TUM datasets [18] consist of 39 sequences recorded in different indoor scenarios using Microsoft Kinect sensor. The dataset comprises various sets of data for distinct tasks, each encompassing multiple data points. These datasets can be utilized to evaluate the performance of multiple tasks. In this paper, $freiburg2\_desk$ sequence and $freiburg2\_large\_with\_loopsequence$ are selected for testing. The average angular velocity and displacement velocity of $freiburg2\_desk$ sequence cameras are $6.338deg/s$, $0.193m/s$, and the total duration is $99.36s$. The average angular velocity of camera rotation in $freiburg2\_large\_with\_loop$ sequence is $17.211deg/s$, the average displacement velocity is $0.231m/s$, and the total duration is $173.19s$. Since the motion of this sequence is fast and the running time is long enough, the effect of the constant acceleration motion model on the system can be fully evaluated.

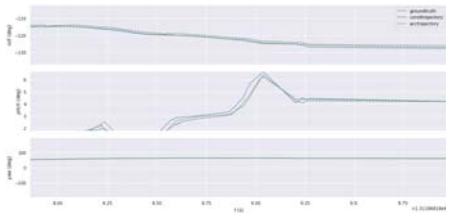


(a) overall trajectory comparison (b) corner trajectory comparison

Fig. 3 Comparison of initial pose results between our method and the original method on $freiburg2\_desk$ sequence



(a) deviation from reference on axis xyz of the world coordinate



(b) deviation from reference in rotation angle of pose

Fig. 4 The deviation of the initial pose of the constant acceleration model and constant velocity model compared with the reference on $freiburg2\_desk$ sequence

TABLE I
COMPARISON OF TRACK ACCURACY ON $freiburg2\_desk$ SEQUENCE

	Original method		proposed method		improvement	
	ATE(m)	RPE(m)	ATE(m)	RPE(m)	ATE(%)	RPE(%)
max	0.041667	0.031183	**0.035482**	**0.026109**	**14.84**	**16.27**
mean	0.019566	0.002914	**0.018048**	**0.002775**	**7.76**	**4.77**
median	0.019975	0.002543	**0.017585**	**0.002458**	**11.96**	**3.34**
min	0.003574	**0.000151**	**0.001619**	0.000250	**54.7**	-
rmse	0.020342	0.003488	**0.018997**	**0.003211**	**6.61**	**7.94**
sse	0.899571	0.026443	**0.784561**	**0.022404**	**12.78**	**15.27**
std	**0.005566**	0.001917	0.005928	**0.001616**	-	**15.70**

TABLE II
COMPARISON OF TRACK ACCURACY ON $freiburg2\_large\_with\_loop$ SEQUENCE

	original method		proposed method		improvement	
	ATE(m)	RPE(m)	ATE(m)	RPE(m)	ATE(%)	RPE(%)
max	0.356589	0.6131805	**0.321118**	**0.509253**	**9.95**	**16.95**
mean	0.132854	0.018648	**0.130115**	0.018656	**2.06**	-
median	0.138500	0.013229	**0.134309**	**0.012865**	**4.48**	**2.75**
min	0.009728	0.000903	**0.005754**	**0.000488**	**40.85**	**45.96**
rmse	0.146425	0.030712	**0.141616**	**0.029711**	**3.28**	**3.23**
sse	25.964133	1.141311	**24.286712**	**1.068115**	**6.46**	**6.41**
std	0.061564	0.024403	**0.055903**	**0.023123**	**9.20**	**5.25**

B. Evaluation Results

We used both Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) to evaluate our proposed method. Specifically, the ATE is:

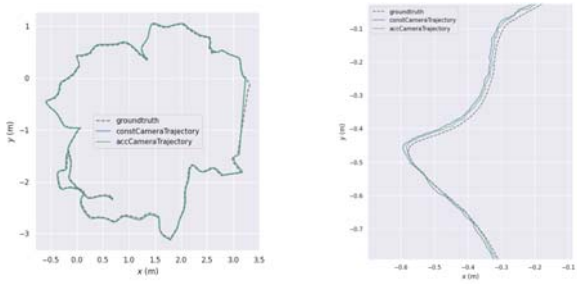
$$ATE_{all} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|log(T_{gt,i}^{-1} T_{esti,i})^{\vee}\|_2^2} \tag{21}$$

The RPE is:

$$RPE_{all} = \sqrt{\frac{1}{N-\triangle t}\sum_{i=1}^{N-\triangle t} \|log(M)^{\vee}\|_2^2} \tag{22}$$

where $M = (T_{gt,i}^{-1} T_{gt,i+\triangle t})^{-1} (T_{esti,i}^{-1} T_{esti,i+\triangle t})$. T_{gt_i} is real trajectory, T_{esti_i} is estimated trajectory.

Fig. 3 shows the initial pose estimation results of our proposed method and the original method of ORBSLAM3 on $freiburg2\_desk$ sequence. Since the actual motion speed of the test sequence is not very fast, there is little difference in the overall trajectory. As can be seen from Fig. 3(b), at the corner, the initial pose estimated by the constant acceleration model is closer to the real trajectory. Fig. 4(a) shows the deviation of the initial pose of the two models compared with the reference on the xyz axis of the world coordinate, and Fig. 4(b) shows the deviation of the rotation angle of the pose compared with the reference. After the analysis of the error estimation in Section III, we know that the motion model needs to provide a more accurate initial pose. It can be seen from the figure that our proposed method has better results. We also show the pose results after optimizing the initial pose provided by the two motion models. As shown in Fig. 5, it can be seen that the optimized trajectory obtained by using our proposed method is closer to the real trajectory. Fig. 6 shows the results of testing on $freiburg2\_large\_with\_loop$ sequence, which is a long sequence and moves fast. Fig. 6(a) shows the comparison between the trajectories generated by the two motion models and the real trajectory.



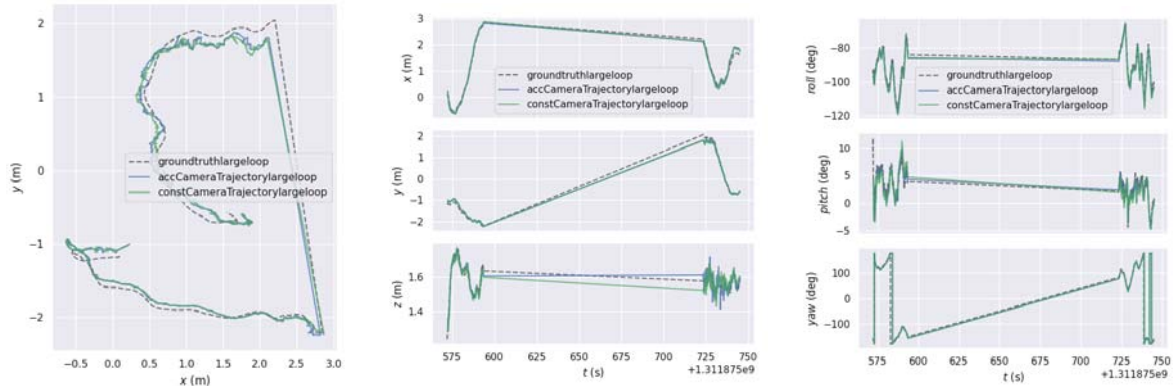
(a) overall trajectory comparison (b) corner trajectory comparison

Fig. 5 Comparison of optimized pose results between our method and the original method on $freiburg2\_desk$ sequence

Table I presents the test results for the $freiburg2\_desk$ sequence, while Table II displays the test results for the $freiburg2\_large\_with\_loop$ sequence. The notation max represents the maximum value, $mean$ represents the mean value, $median$ represents the median value, min represents the minimum value, $rmse$ represents the root mean square error, sse represents the sum of squared residual, and std represents the standard deviation. As evident from the comparison of results presented in Tables I and II, our proposed method demonstrates superiority over the method employed by ORBSLAM3 in nearly all performance indicators. In particular, the $rmse$ of ATE is improved by 6.61% on the $freiburg2\_desk$ sequence and improved by 6.46% on the $freiburg2\_large\_with\_loop$ sequence compared with the original method.

V. CONCLUSION

In this paper, we proposed a motion tracking model. We assume that the acceleration of camera motion remains constant over a brief time interval. At the same time, we decoupled the transformation matrix and estimated the rotation matrix and the translation vector respectively. Furthermore, a



(a) trajectory comparison (b) deviation from reference on axis xyz (c) deviation from reference in rotation angle of pose

Fig. 6 Comparison of the results evaluated by the constant acceleration model and the constant velocity model on $freiburg2\_large\_with\_loop$ sequence

theoretical analysis was conducted to examine the tracking error of the constant velocity model. The findings indicate that the error is influenced by variations in both angular velocity and displacement vector. Finally, the proposed method was implemented and subsequently evaluated using the TUM dataset. Our methodology enhances the precision of the ORBSLAM3 system and offers a comparatively more precise estimation of the initial pose. In our forthcoming research, we intend to incorporate the motion model into the ORBSLAM3 system during the key frame extraction process, with the aim of enhancing the overall system performance.

ACKNOWLEDGMENT

This work was supported by Stability Program of Science and Technology on Communication Security Laboratory (2022) and Sichuan Science and Technology Program(No.2021YJ0082).

REFERENCES

[1] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual slam algorithms: A survey from 2010 to 2016," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, pp. 1–11, 2017.

[2] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[3] S. Zheng, J. Wang, C. Rizos, and A. El-Mowafy, "Simultaneous localization and mapping (slam) for autonomous driving," in *IGNSS Conference*, 2020.

[4] 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.

[5] K. Fujii, "Extended kalman filter," *Refernce Manual*, vol. 14, 2013.

[6] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM international symposium on mixed and augmented reality*. IEEE, 2007, pp. 225–234.

[7] D. G. Viswanathan, "Features from accelerated segment test (fast)," in *Proceedings of the 10th workshop on image analysis for multimedia interactive services, London, UK*, 2009, pp. 6–8.

[8] B. Triggs, A. Zisserman, and R. Szeliski, *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*. Springer, 2000.

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

[10] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[11] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *2011 international conference on computer vision*. IEEE, 2011, pp. 2320–2327.

[12] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE international symposium on mixed and augmented reality*. Ieee, 2011, pp. 127–136.

[13] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 15–22.

[14] S. Y. Loo, A. J. Amiri, S. Mashohor, S. H. Tang, and H. Zhang, "Cnn-svo: Improving the mapping in semi-direct visual odometry using single-image depth prediction," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019.

[15] R. Elvira, J. D. Tardós, and J. Montiel, "Orbslam-atlas: a robust and accurate multi-map system," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 6253–6259.

[16] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 834–849.

[17] M. Grupp, "evo: Python package for the evaluation of odometry and slam." https://github.com/MichaelGrupp/evo, 2017.

[18] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 573–580.

Daohong Yang received the bachelor of communication engineering from Donghua university in 2021. He is currently a master's student at the University of Electronic Science and Technology of China. His research focuses on visual SLAM.

Xiang Zhang received the bachelor of communication engineering from University of Electronic Science and Technology of China in 2020 and received the his master's degree from University of Electronic Science and Technology of China in 2023. His research interest covers Visual SLAM and integrated circuits.

Wanting Zhou received the Ph.D. degree in communication and information systems from University of Electronic Science and Technology of China, Chengdu, China, in 2014. She is currently a research associate with Universityof Electronic Science and Technology of China. Her research interests include integrated circuits, machine Learning and hardware security.

Lei Li received Ph.D. degree in communication engineering from University of Electronic Science and Technology of China, Chengdu, China, in 2010. He is currently an associate researcher with University of Electronic Science and Technology of China. His current research interests include integrated circuits, machine learning, and hardware security.