J. Meas. Sci. Instrum., 2026, 17(1): 36-48.
DOI: 10.62756/jmsi.1674-8042.2026003
http://jmsi.publish.founderss.cn
jmsi@nuc.edu.cn

Enhancing monocular visual positioning: 3D target-assisted initialization

MA Junyu, SUN Changku, WANG Peng*, FU Luhua

State Key Lab of Precision Measurement Technology and Instruments, Tianjin University, Tianjin 300072, China

***Corresponding author:** WANG Peng (wang_peng@tju.edu.cn)

Received: March 3, 2025 **Revised:** April 5, 2025 **Accepted:** April 10, 2025

Abstract: Monocular visual positioning systems are valued for their low cost and straightforward calibration. However, the lack of real-scale information and the complexity of initialization processes limit their application in scenarios requiring accurate absolute positioning. Existing solutions present trade-offs: markerless approaches depend on environmental priors (e.g., fixed camera height constraints) for scale recovery, while marker-based methods typically necessitate that target patterns remain within the camera's field of view throughout the process. To address these challenges, we propose a 3D target-assisted initialization method that enables scale recovery with just two target images. This modular approach can be seamlessly integrated into monocular simultaneous localization and mapping (SLAM) frameworks. We validated our proposed initialization method through integration with ORB-SLAM3 and semi-direct visual odometry (SVO). Experimental results demonstrated that our method provides real-scale information without compromising real-time performance, making it suitable for applications such as indoor navigation and industrial robot localization, where accurate absolute positioning is essential.

Key words: monocular simultaneous localization and mapping (SLAM); real-time positioning; scale recovery; 3D target; absolute positioning; visual odometry

0 Introduction

Positioning technology serves as the foundation for all location-based services. For outdoor positioning, the global navigation satellite system (GNSS) is the primary solution due to its technical maturity, high accuracy, and real-time performance, achieving sub-meter-level positioning accuracy. However, GNSS signals experience significant attenuation in indoor environments, rendering GNSS ineffective for indoor applications[1-2].

Indoor positioning systems have a wide range of applications, including navigation in shopping malls, worker dispatching and command in factories, emergency services in public indoor spaces, and vehicle guidance in large underground parking. To meet the demands of indoor positioning, various technologies have emerged. ZigBee-based indoor positioning technology achieves localization through wireless electromagnetic communication between central nodes, gateway devices, and blind nodes, offering strong scalability. However, it is highly susceptible to multipath effects caused by indoor obstacles, which significantly hinder its signal propagation[3]. Wi-Fi-based indoor positioning technology relies on multiple Wi-Fi signal nodes to determine location based on received signal strength. This method has a low deployment cost and meets general accuracy requirements; however, its signals are highly susceptible to environmental interference[4]. Bluetooth indoor positioning technology uses deployed Bluetooth signal transmitters to determine location based on signal strength, offering low power consumption but limited transmission range[5]. Ultra-wideband (UWB) indoor positioning technology can achieve sub-meter-level accuracy but requires the installation of dedicated anchors, which results in high costs and inconvenient deployment[6]. Laser tracking instruments can achieve millimeter-level positioning accuracy but can only track one target at a time[7]. Above all, the laser-based measurement method is infeasible in many application scenarios, limiting its adoption.

In recent years, visual simultaneous localization and mapping (SLAM) has emerged as a practical indoor positioning approach, gaining attention for its independence from external signal sources, low cost, and capability to ensure high positioning accuracy and speed[8]. Among existing visual SLAM technologies, stereo SLAM can provide depth information through calibrated baselines but suffers from the drawbacks of complex calibration processes and high computational cost[9]. RGB-D SLAM, while

© The Author(s) 2026. Published by Publishing Center of North University of China. The articles published in this open access journal are distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

capable of directly acquiring depth information, comes with high device costs and is prone to errors under lighting variations[10]. In contrast, monocular SLAM offers the advantages of simple calibration and low cost[11]. However, conventional monocular SLAM faces two major challenges. On one hand, it lacks real-scale information, meaning that its measurements can only provide relative coordinates, which fail to meet the requirements for absolute positioning. On the other hand, the initialization of monocular SLAM relies on feature matching between images, making the process complex and less stable in dynamic environments[12].

As an end-to-end framework, deep learning has been introduced to address the above issues by directly estimating positioning results from monocular input. Deep learning approaches can be divided into unsupervised and supervised methods. Based on unsupervised learning, Li et al. proposed a monocular visual odometry system called UnDeepVO, which uses stereo images for training to recover scale information and applies monocular images during operation. Compared to methods without deep learning or those trained with monocular images, this approach achieves real-scale recovery[13]. For supervised learning, Kim et al. proposed a neural network-based semantic visual odometry architecture that integrates visual odometry, object detection, and instance segmentation. Additionally, they introduced an attention-based pose estimation architecture that employs multitask learning to handle dynamic objects and improve performance[14].

However, there are also some challenges with deep learning approaches. On one hand, training deep learning models typically requires a large amount of data, and obtaining high-quality labeled data is often a complex and costly process. On the other hand, deep learning models lack interpretability, making the decision-making processes difficult to understand and posing risks in application scenarios with high reliability requirements. Furthermore, the performance of deep learning models is often influenced by data distribution, resulting in significant performance variations across different environments[15-16].

Another approach utilizes multi-sensor fusion frameworks to achieve absolute positioning by combining inertial measurement unit (IMU) data with visual information. However, such methods increase hardware complexity and raise system deployment costs. In conclusion, monocular SLAM still lacks a cost-effective and easily implementable solution for providing real-scale information.

Several other approaches have been proposed for monocular scale recovery, some of which do not rely on deep learning and only use monocular vision. One such approach involves aggregating planar features from the environment and fitting a ground plane to estimate the scale. This approach assumes a constant height of the camera above the ground and relies on robust extraction and aggregation of ground points, which is followed by solving for the scale using a least-squares optimization. While this technique can be effective in specific applications, the camera's height must remain constant, which limits its usability in dynamic environments where the camera's position or orientation changes, such as in wearable SLAM systems[17].

Another approach involves using coded fiducial markers placed within the environment. By decoding the coordinates of known targets through the camera's images, this approach provides scale information with high precision. However, it faces a significant limitation: when the targets are outside the camera's field of view, scale recovery becomes impossible, making it less robust in unstructured or moving environments[18].

These existing approaches, which offer practical solutions under certain conditions, served as important sources of inspiration for our proposed method. In contrast to the above approaches, we propose a 3D target-based initialization method that utilizes only two images of the target during initialization. This method streamlines the monocular SLAM pipeline, enabling real-scale recovery without the need for multi-sensor fusion or graphics processing unit (GPU)-accelerated computing devices. By removing the reliance on other prior constraints, our method provides a flexible and modular solution that can be integrated with various visual SLAM frameworks. The proposed approach is particularly well-suited for indoor scenarios requiring absolute positioning and can be deployed in a variety of environments without the need for complex setups or external sensors.

ORB-SLAM3 is capable of performing metric-unaware monocular SLAM[19]. Therefore, the initialization method proposed in this paper is integrated into ORB-SLAM3 to validate its feasibility using real-world data.

1 ORB-SLAM3 process

As the application platform for the algorithm proposed in this paper, ORB-SLAM3 consists of three concurrent subsystems: tracking, local mapping, and loop closure detection, which are executed in parallel during the main program's runtime. This parallel architecture is designed to ensure real-time performance while achieving high-precision camera pose estimation and scene map

construction. ORB-SLAM3 is selected as the experimental platform in this study due to its modular thread design, which facilitates the integration of the proposed initialization algorithm and provides a standardized environment for accuracy evaluation consistent with mainstream methods. The image frame processing flow is illustrated in Fig.1.

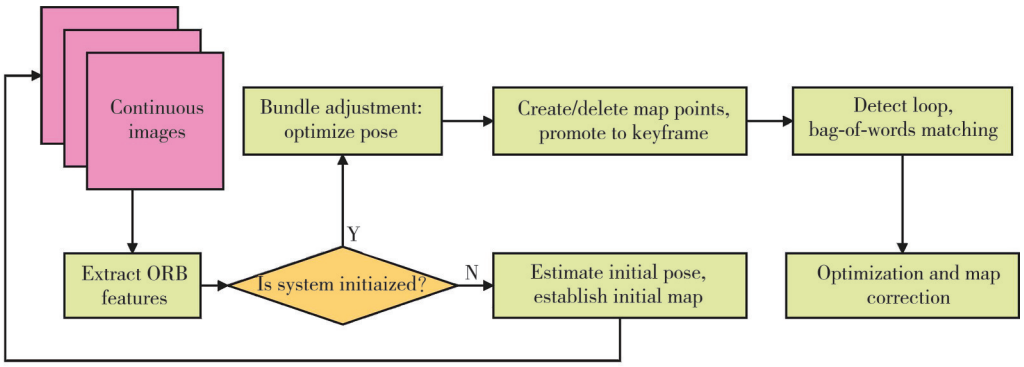


Fig. 1 Processing flow of ORB-SLAM3

Sequential monocular image frames are first fed into the tracking thread. The tracking thread extracts oriented FAST and rotated BRIEF (ORB) feature points from each frame. During its initialization phase, the thread uses epipolar geometry to estimate the initial pose from two temporally close frames, and the two frames that successfully recover the initial pose and establish the initial map point cloud are designated as keyframes. Subsequently, the pose of each image frame is optimized based on the nearest preceding keyframe in time, using a constant-motion model and bundle adjustment. This optimization strategy dynamically adjusts pose estimation results to adapt to complex scenarios. However, the accuracy of the pose estimation is partially dependent on the accuracy of the initialization phase.

Based on the initial map point cloud obtained during initialization, the local mapping thread processes subsequent input frames. Its main task is to determine whether to promote the current frame to a new keyframe based on factors such as the quality of the current frame, the time interval from the last keyframe, and the history of relocalization. This dynamic keyframe selection mechanism allows the system to adapt to environmental changes while avoiding the generation of redundant keyframes, thereby reducing computational costs. The creation of new keyframes introduces feature matching with existing keyframes, establishes new covisibility constraints, and generates or removes map points to optimize the structure and accuracy of the map.

While the tracking thread processes new image frames, the loop closure detection thread is responsible for identifying loop closures in the trajectory of the image frames. This is achieved by using a bag-of-words model to detect and match candidate keyframes for loop closure. When the bag-of-words vector of the current image frame matches a previously visited keyframe at a level exceeding a predefined threshold, a loop closure is identified. When a loop closure is detected, similarity transformation groups and bundle adjustment optimization will be used to perform error correction on the map. This process eliminates accumulated errors and improves the global consistency of the map. In addition, during the execution of the algorithm, the co-visibility graph and essential graph are continuously optimized and updated. The co-visibility graph records the co-visibility relationships between keyframes, and the essential graph captures high-weight co-visibility relationships. These two graph structures play a critical role in expanding the search range for the local map during tracking, creating new map points between keyframes in local mapping, and supporting loop closure detection and relocalization.

2 3D target-assisted initialization method

A monocular SLAM system consists of a camera and its host computer software. The camera captures environmental images, and the host software calculates the real-time camera pose. Before initialization, the camera's intrinsic matrix and distortion parameters are obtained using the Zhang's calibration method as known parameters.

2.1 Design of 3D target

The design of the target aims to enable the acquisition of the camera optical center's pose relative to the world coordinate system using only a single image frame of the target. If a planar target was used, all obtained feature points would be coplanar, so the depth information in 3D space would be correlated, making it impossible to determine the camera pose using only a single image frame. Therefore, the target should include multiple feature points with known world coordinates that are not coplanar. Additionally, these feature points should be easily

detectable with minimal computational effort.

The 3D target designed in this method is shown in Fig.2. The target consists of a cube, with each of its six faces covered by a distinct AprilTag[20], each with a unique ID. The use of AprilTag allows for fast recognition and decoding, ensuring real-time performance in the initialization algorithm. Each face's AprilTag provides four feature points with directional positioning information, which remains stable regardless of tag rotation. These feature points are used to determine the target's pixel coordinates based on the unique ID of the AprilTag.

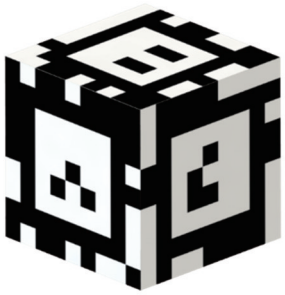


Fig. 2 3D target

During the initialization process, three adjacent complete AprilTag target images are captured, forming three distinct information planes. Each information plane contains four feature points, resulting in a total of twelve feature points that are not coplanar in 3D space. The non-coplanar nature of these feature points is essential for overcoming the depth correlation issues present in planar targets, enabling accurate camera pose estimation from a single image frame.

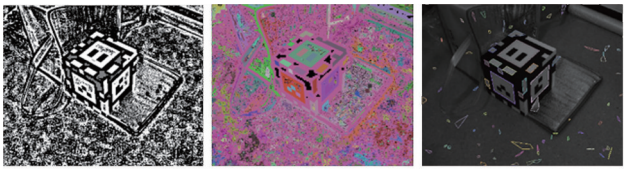
Given that the world coordinates of the feature points are known, the pixel coordinates of the feature points in two images captured by the camera are used to compute the rotation matrix and translation vector between the two frames. The extracted keypoints are then triangulated to generate a 3D point cloud, which is used to perform the initialization while incorporating scale information.

2.2 Feature extraction of 3D target

After the camera captures an image, the image is processed in following four stages to finish extraction of feature points.

2.2.1 Binarization

The image captured by the camera is a single-channel grayscale image with a resolution of 1 280×1 024 pixels. For computational efficiency, the image is downsampled to 640×512 pixels. We perform binarization using an adaptive thresholding method, which processes blocks of 5×5 pixels to enhance edge detection. Regions with little edge information are excluded from further processing to reduce computational load, as shown in Fig.3(a).



(a) Binarization (b) Connected component extraction (c) All candidate quadrilaterals

Fig. 3 Pre-decoding process

2.2.2 Identifying connected components

The binarized image includes both the target's edges and some false detections. Using the union-find algorithm, we extract edge pixels that could form the target's boundary contours. The result segments the image into connected components, with quadrilateral regions representing the target's localization quadrilaterals, as shown in Fig.3(b).

2.2.3 Fitting quadrilateral target contour

A contour fitting method is employed to identify the quadrilateral region corresponding to the target. The boundary points of each connected component are first sorted in a consistent clockwise order based on their angular positions relative to the centroid. Corner points are detected by analyzing local variations along the boundary. Line segments are fitted to subsets of boundary points using Principal Component Analysis, which computes the principal components of the point distributions. The primary eigenvector of the covariance matrix represents the best-fit line, and deviations in the mean square error are used to identify candidate corner points. For each candidate quadrilateral, fitting errors are calculated, and those with excessive fitting errors, fewer than four valid corner points, or angles deviating significantly from 90° are discarded. Quadrilaterals that meet these criteria are retained for further decoding, as shown in Fig.3(c).

2.2.4 Decoding

After obtaining the quadrilateral contours, the data bits within the contour are decoded. If a quadrilateral contour is decoded correctly, it is considered a valid tag, as illustrated in Fig.4. The first step involves perspective correction to recover the true encoded image of the tag. The corrected tag image is then divided into a 9×9 grid of square regions, with bilinear interpolation used to extract the characteristic grayscale value for each region, forming a 9×9 feature matrix. This matrix is subsequently sharpened using a Laplacian kernel, producing the feature matrix for decoding.

To enhance decoding efficiency, all possible orientations of the feature matrix are precomputed and stored in a hash table, which eliminates the need for multiple computations of the Hamming distance from the standard values,

enabling fast decoding with a single query. The decoding process also filters out incorrect candidate quadrilaterals, ultimately yielding the quadrilateral contours of all valid tags and their corresponding tag IDs.

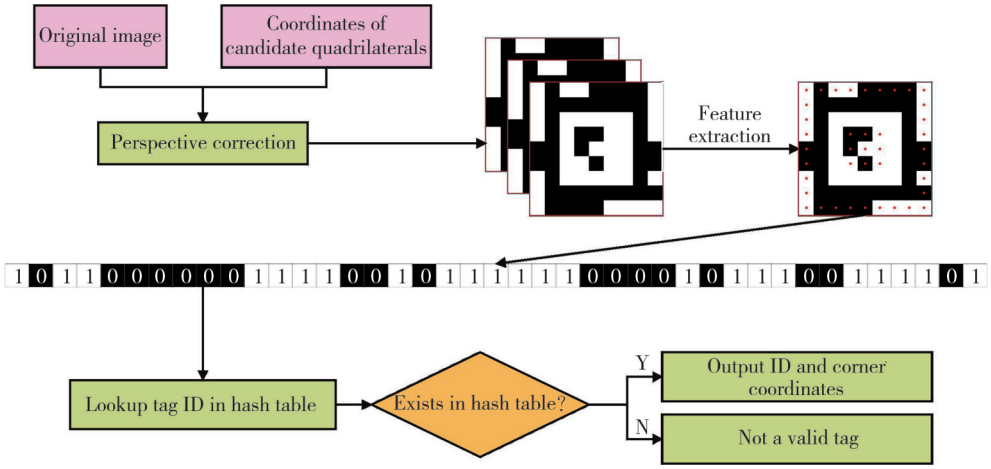


Fig. 4 Illustration of decoding process

2.3 Pose estimation algorithm based on feature points

The above process enables the acquisition of the tag's ID and the pixel coordinates of its feature corners. Additionally, the world coordinates of the target's feature points can be determined based on the ID, thus forming twelve matching pairs of pixel coordinates (u_i, v_i) and world coordinates $\boldsymbol{p}_i^{\mathrm{w}} = (X_i, Y_i, Z_i)$. This leads to a perspective-n-point (PnP) problem, where the rotation matrix and translation vector can be computed through matching pairs. The transformation between pixel coordinates and world coordinates can be represented by the pinhole camera model as

$$z_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \boldsymbol{K} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} = \begin{bmatrix} f_{\mathrm{u}} & 0 & u_{\mathrm{c}} \\ 0 & f_{\mathrm{v}} & v_{\mathrm{c}} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}, \quad (1)$$

where z_i is the scale factor, $\boldsymbol{K}$ is the camera's intrinsic matrix, which has been previously calibrated using Zhang's method, and $\boldsymbol{R}$ and $\boldsymbol{t}$ are the unknown rotation and translation matrices to be solved.

In the PnP problem, if the 2D-3D matching point set contains a certain proportion of incorrect matches, we can employ a random sample consensus method to reduce the effect of incorrect matches. In this method, n points are randomly selected from all the points to compute the rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$. By repeating this process multiple times, the most consistent result is obtained, which represents the camera's pose. However, since the matching point pairs determined by the 3D target have already undergone correctness verification, no incorrect matches occur. Therefore, all 12 points are directly incorporated into the calculation. We adopt the EPnP algorithm to compute the rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$.

All 3D points are expressed as a weighted sum of four virtual control points $\boldsymbol{c}_0^{\mathrm{w}}$, $\boldsymbol{c}_1^{\mathrm{w}}$, $\boldsymbol{c}_2^{\mathrm{w}}$, $\boldsymbol{c}_3^{\mathrm{w}}$, thereby reducing the dimensionality of the original problem and avoiding direct handling of high-dimensional point clouds by

$$\boldsymbol{p}_i^{\mathrm{w}} = \sum_{j=0}^{3} \alpha_{ij} \boldsymbol{c}_j^{\mathrm{w}}, \quad i = 1, 2, \cdots, n, \quad (2)$$

where α_{ij} are the weight coefficients, satisfying the following condition as

$$\sum_{j=0}^{3} \alpha_{ij} = 1. \quad (3)$$

The first virtual control point $\boldsymbol{c}_0^{\mathrm{w}}$ can be calculated directly by

$$\boldsymbol{c}_0^{\mathrm{w}} = \frac{1}{N+1} \sum_{i=0}^{N} \boldsymbol{p}_i^{\mathrm{w}}. \quad (4)$$

From this, we can compute the de-centered 3D points and construct the matrix $\boldsymbol{W}_{\mathrm{c}}$ as

$$\boldsymbol{p}_i^{\mathrm{wo}} = \boldsymbol{p}_i^{\mathrm{w}} - \boldsymbol{c}_0^{\mathrm{w}}, \quad (5)$$

$$\boldsymbol{P}^{\mathrm{wo}} = [\, \boldsymbol{p}_1^{\mathrm{wo}} \cdots \boldsymbol{p}_i^{\mathrm{wo}} \cdots \boldsymbol{p}_N^{\mathrm{wo}} \,]_{3 \times n}, \quad (6)$$

$$\boldsymbol{W}_{\mathrm{c}} = \boldsymbol{P}^{\mathrm{wo}} \boldsymbol{P}^{\mathrm{woT}}. \quad (7)$$

Then, performing singular value decomposition (SVD) on $\boldsymbol{W}_{\mathrm{c}}$, namely

$$\boldsymbol{W}_{\mathrm{c}} = \boldsymbol{U}_{\mathrm{c}} \boldsymbol{\Sigma}_{\mathrm{c}} \boldsymbol{V}_{\mathrm{c}}^{\mathrm{T}}, \quad (8)$$

we can extract the direction information related to the control point distribution from the singular values σ_1, σ_2, σ_3 and the column vectors $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, $\boldsymbol{v}_3$, which can be used to directly obtain the coordinates of the other three control points as

$$\boldsymbol{c}_j^{\mathrm{w}} = \boldsymbol{c}_0^{\mathrm{w}} + \sqrt{\frac{\sigma_j}{N+1}}\, \boldsymbol{v}_j, \ j \in [1, 3]. \quad (9)$$

By constructing the matrixs

$$\begin{bmatrix} \boldsymbol{p}_i^{\mathrm{w}} \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{c}_0^{\mathrm{w}} & \boldsymbol{c}_1^{\mathrm{w}} & \boldsymbol{c}_2^{\mathrm{w}} & \boldsymbol{c}_3^{\mathrm{w}} \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_{i0} \\ \alpha_{i1} \\ \alpha_{i2} \\ \alpha_{i3} \end{bmatrix}, \tag{10}$$

$$\begin{bmatrix} \alpha_{01} & \alpha_{11} & \dots & \alpha_{N1} \\ \alpha_{02} & \alpha_{12} & \dots & \alpha_{N2} \\ \alpha_{03} & \alpha_{13} & \dots & \alpha_{N3} \end{bmatrix} = \boldsymbol{C}^{-1} \boldsymbol{P}^{\mathrm{wo}}, \tag{11}$$

we then combine Eq. (3) and Eq. (11) to obtain the control point weight coefficients for all world points, and the control point representation of all world points is derived. Using the camera's intrinsic matrix, weight coefficients, 3D point coordinates, and corresponding 2D image points, we establish the projection model equation and solve for the coordinates of the four control points in the camera coordinate system. This is formulated as

$$\boldsymbol{p}_i^{\mathrm{c}} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_i^{\mathrm{w}} \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \end{bmatrix} \begin{bmatrix} \sum_{j=0}^{3} \alpha_{ij} \boldsymbol{c}_j^{\mathrm{w}} \\ \sum_{j=0}^{3} \alpha_{ij} \end{bmatrix} =$$

$$\sum_{j=0}^{3} \alpha_{ij} \left(\boldsymbol{R} \boldsymbol{c}_j^{\mathrm{w}} + \boldsymbol{t} \right) = \sum_{j=0}^{3} \alpha_{ij} \boldsymbol{c}_j^{\mathrm{c}} = \sum_{j=0}^{3} \alpha_{ij} \begin{bmatrix} x_j^{\mathrm{c}} \\ y_j^{\mathrm{c}} \\ z_j^{\mathrm{c}} \end{bmatrix}, \tag{12}$$

$$\boldsymbol{p}_i^{\mathrm{c}} = \boldsymbol{K}^{-1} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} z_i. \tag{13}$$

By combining Eq. (12) and Eq. (13), we derive

$$\begin{cases} \sum_{j=0}^{3} \alpha_{ij} f_{\mathrm{u}} x_j^{\mathrm{c}} + \alpha_{ij} \left(u_{\mathrm{c}} - u_i \right) z_j^{\mathrm{c}} = 0, \\ \sum_{j=0}^{3} \alpha_{ij} f_{\mathrm{v}} y_j^{\mathrm{c}} + \alpha_{ij} \left(v_{\mathrm{c}} - v_j \right) z_j^{\mathrm{c}} = 0. \end{cases} \tag{14}$$

Since there are 12 world points, we obtain a system of linear equations with 24 equations based on the above formulas. Solving for the control points' coordinates and weights in the world coordinate system relative to the camera coordinate system means all 3D points' coordinate marks are considered. After obtaining the 3D-3D point pairs, we perform a singular value decomposition to derive the rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$.

Initially, the camera coordinate system is decentered by the geometric transformation based on the following method

$$\boldsymbol{p}_i^{\mathrm{co}} = \boldsymbol{p}_i^{\mathrm{c}} - \frac{1}{N+1} \sum_{i=0}^{N} \boldsymbol{p}_i^{\mathrm{c}}, \tag{15}$$

$$\boldsymbol{P}^{\mathrm{co}} = \left[\boldsymbol{p}_1^{\mathrm{co}} \cdots \boldsymbol{p}_i^{\mathrm{co}} \cdots \boldsymbol{p}_N^{\mathrm{co}} \right]_{3 \times n}, \tag{16}$$

$$\boldsymbol{W} = \boldsymbol{P}^{\mathrm{co}} \boldsymbol{P}^{\mathrm{wo\,T}}. \tag{17}$$

Performing SVD on the matrix $\boldsymbol{W}$ gives

$$\boldsymbol{W} = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^{\mathrm{T}}. \tag{18}$$

Thus, we can get

$$\boldsymbol{R} = \boldsymbol{U} \boldsymbol{V}, \quad \boldsymbol{t} = \frac{1}{N+1} \sum_{i=0}^{N} \boldsymbol{p}_i^{\mathrm{c}} - \boldsymbol{R} \boldsymbol{c}_0^{\mathrm{w}}. \tag{19}$$

Finally, the rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$ from the world coordinate system to the camera coordinate system are obtained from the target pattern.

2.4 Initialization process

At this point, the current camera pose can be recovered from a single image of the complete 3D target pattern, and the system is ready for initialization. The algorithm is summarized in Fig.5.

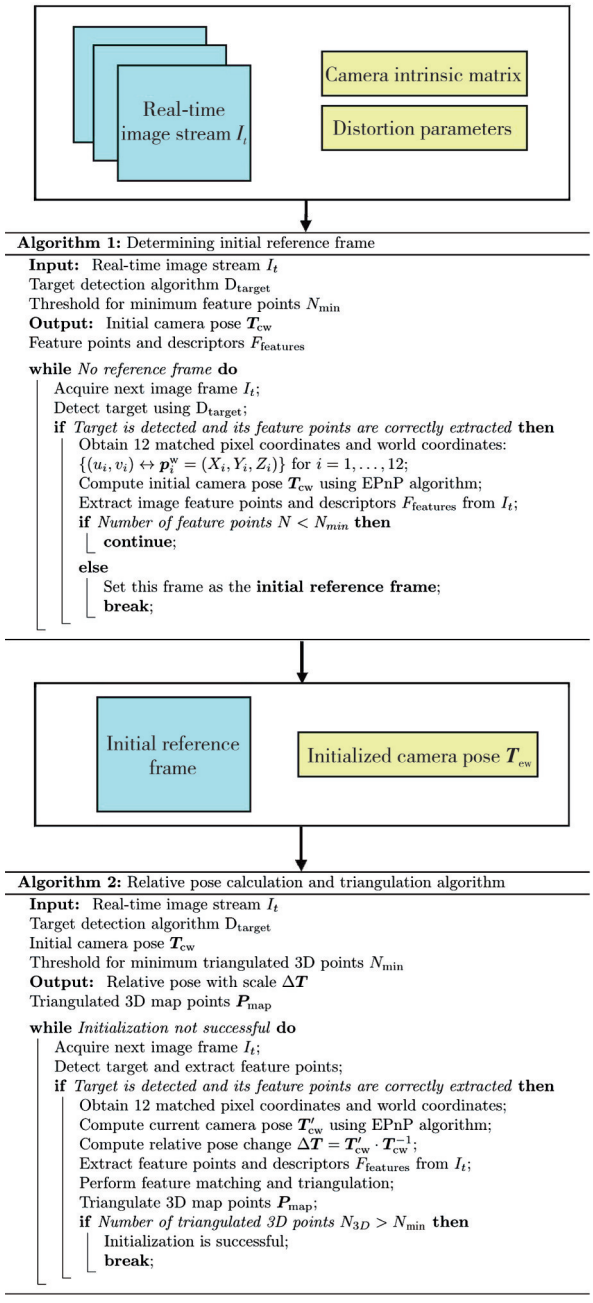


Fig. 5 System initialization algorithm

Given known camera intrinsics and distortion coefficients, sequential image frames are acquired from the camera. If the pose of a frame is computed using the 3D target and enough feature points are obtained for further

processing in ORB-SLAM3, that frame is considered the initial reference frame. After obtaining the initial reference frame, the following frames are checked to determine if both conditions are met: the pose has been determined with the 3D target, and the number of sparse 3D points triangulated with the initial reference frame exceeds a predefined threshold. Once these criteria are satisfied, the initial mapping is completed, and two reference frames containing scale information are obtained, thus finalizing the system's initialization. Afterward, continuous frames can be fed into the system to achieve real-time determination of the camera's position in the established reference coordinate system.

2.5 Camera pose estimation algorithm for enhancing initialization robustness

The design of the 3D target provides enough geometric constraints for high-precision camera pose estimation. However, in practical measurements, environmental conditions and camera angles may limit the complete extraction of feature points.

When the camera is at extreme angles, it is difficult to obtain complete views of all three information planes due to restricted perspectives, resulting in the inability to extract all the 12 feature points, which is shown in Fig. 6(a) that only two information planes can be successfully extracted.

(a) Extreme viewing angle (b) Extreme illumination

Fig. 6 Missing target feature points in extreme conditions

Another scenario arises in high dynamic range environments or conditions with drastic lighting variations, leading to overexposure or underexposure. For example, even when all three information planes of the 3D target are in the image frame, the camera's limited dynamic range makes the detection of feature points from all three planes fail, as shown in Fig.6(b).

To solve the above issues, we propose a camera pose estimation algorithm for enhancing initialization robustness, as shown in Fig. 7. We adjust the pose estimation strategy through a flexible mechanism for judging the number of detected feature points when the information planes of the 3D target are incompletely recognized.

Input: Image frame I
Camera intrinsic parameters $\boldsymbol{K}$
Predefined 3D target model with known feature points $\{\boldsymbol{P}_i^{\mathrm{w}}\}$
Output: Estimated camera pose $\boldsymbol{T}_{\mathrm{cw}}$

Step 1: Extract feature points from 3D target
Detect 3D target in the image frame I and extract feature points $\{(u_i, v_i)\}$.

Step 2: Evaluate information planes
if *Three information planes are successfully detected* **then**
  Use all feature points $\{(u_i, v_i) \leftrightarrow \boldsymbol{P}_i^{\mathrm{w}}\}$ for pose estimation;
  Compute $\boldsymbol{T}_{\mathrm{cw}}$ using EPnP;
  return $\boldsymbol{T}_{\mathrm{cw}}$;
else if *Two information planes are successfully detected* **then**
  Use partial feature points $\{(u_i, v_i) \leftrightarrow \boldsymbol{P}_i^{\mathrm{w}}\}$ for pose estimation;
  Compute $\boldsymbol{T}_{\mathrm{cw}}$ with reduced accuracy;
  return $\boldsymbol{T}_{\mathrm{cw}}$;
else if *Only one information plane is detected* **then**
  Insufficient geometric constraints for pose estimation;
  Wait for the next frame I_{t+1};
  return "Wait for next frame";

Step 3: Finalize pose estimation
Once sufficient feature points are detected and $\boldsymbol{T}_{\mathrm{cw}}$ is computed, proceed to the next initialization step.

Fig. 7 Camera pose estimation algorithm for enhancing initialization robustness

When all three information planes are extracted, we use the complete set of feature points for pose estimation, achieving the highest accuracy. When two information planes are extracted, the algorithm utilizes partial feature points to estimate the pose; although the accuracy is slightly reduced, it still meets the initialization requirements. When only one information plane is extracted, the insufficient geometric constraints cause pose estimation to degrade. In this case, the algorithm skips the current frame and waits to process the next image frame.

Compared to using three information planes, utilizing two information planes significantly expands the effective range for target pose estimation. The effective range is extended to regions where at least two information planes can be detected, demonstrating that the aforementioned extreme angle issue can be effectively resolved. Under extreme lighting conditions, the algorithm ensures that within the detectable range of the three information planes, it can still handle situations where one plane is missing due to lighting issues.

This algorithm significantly improves the success rate of camera pose estimation under complex angles and lighting conditions. Moreover, through a reasonable mechanism for evaluating the number of feature points, the algorithm effectively avoids unnecessary computational overhead, ensuring the real-time performance and efficiency of the system.

However, the reduction in the number of feature points will lead to instability in the computed pose results. Therefore, this method serves as an optional enhancement for monocular SLAM positioning tasks in complex environments but should not be universally applied to general cases.

3 Experiment and analysis

3.1 Experimental setup

The experimental equipment used in this study included a Hikvision MV-CA013-21UM grayscale camera equipped with an 8 mm focal length lens. The camera operated in internal trigger mode to continuously capture grayscale images at a fixed frame rate with a resolution of 1 280×1 024 pixels. The 3D target was a cube with dimensions of 18 cm×18 cm×18 cm, fabricated from frosted plastic to minimize mirror reflections, with AprilTag patterns printed directly on its surface to ensure high visibility and stable feature detection under various lighting conditions. The experimental environment was configured on a system running Ubuntu 20.04 with an x86_64 architecture. The CPU used was an Intel Core i5-10300H with 16 GB of RAM. The initialization algorithm based on the 3D target was implemented using the C++17 standard.

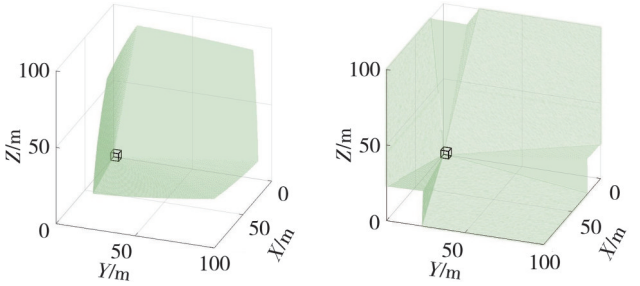
3.2 Experiment on effective initialization range

The effective range of 3D target initialization determines the flexibility of the camera during the initialization phase. If the effective angular range is too narrow, it imposes strict restrictions on the position of the camera, thereby reducing the practicality of the algorithm.

The effective range of the 3D target depends on the range within which the target surface information can be correctly extracted. Experimental validation showed that the critical angle between the target surface and the imaging plane for correct extraction was 77.16°.

Based on the obtained critical angle, the effective range of the camera's position during initialization is shown in Fig. 8. When using the basic pose estimation algorithm that computes the pose using three information planes, the effective range is represented by the green region in Fig. 8(a). When the pose estimation method with enhanced initialization robustness is applied, the effective range expands as shown in Fig.8(b). The blank areas in Fig.8(b) indicate regions where the camera can detect at most one information plane, making it impossible to get the camera's pose.

The experimental results showed that the initialization algorithm proposed in this paper provided a large effective range, thus avoiding overly stringent requirements on the camera's position. Furthermore, the camera pose estimation algorithm with enhancing initialization robustness further expands the effective range, significantly improving the stability of the initialization.



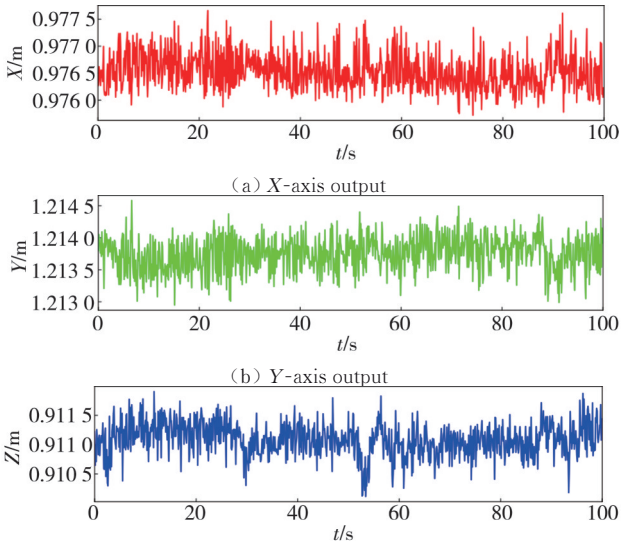
(a) Basic pose estimation (b) Enhanced pose estimation

Fig. 8 Effective range of camera positions during initialization

3.3 Single-frame camera pose estimation experiment based on 3D target

The accuracy of the camera pose obtained from the 3D target is a critical factor influencing the scale accuracy provided by the initialization algorithm. Therefore, this experiment aimed to verify the stability of data acquisition when the camera pose remained fixed and the accuracy of the obtained pose at multiple sampling points.

The camera was kept stationary, and the basic algorithm, which computes the pose using three information planes, was applied to obtain the camera pose outputs continuously for 100 s. The results are shown in Fig.9.



(a) *X*-axis output (b) *Y*-axis output (c) *Z*-axis output

Fig. 9 Three-axis coordinate outputs in static condition

The stability of the data was evaluated using the root mean square error (RMSE), and the results are shown in Table 1.

Table 1 RMSE in static condition

Axis	RMSE/cm
X	0.033 36
Y	0.026 40
Z	0.028 01

Furthermore, the enhanced algorithm was applied to evaluate its ability to estimate poses when the target

extraction was incomplete. Interference was introduced so that only two information planes could be detected per frame. We continuously varied the interference to ensure that the two successfully extracted information planes were not always the same but two random planes out of the three. The camera pose outputs were recorded continuously for 100 s, and the results are shown in Fig.10.

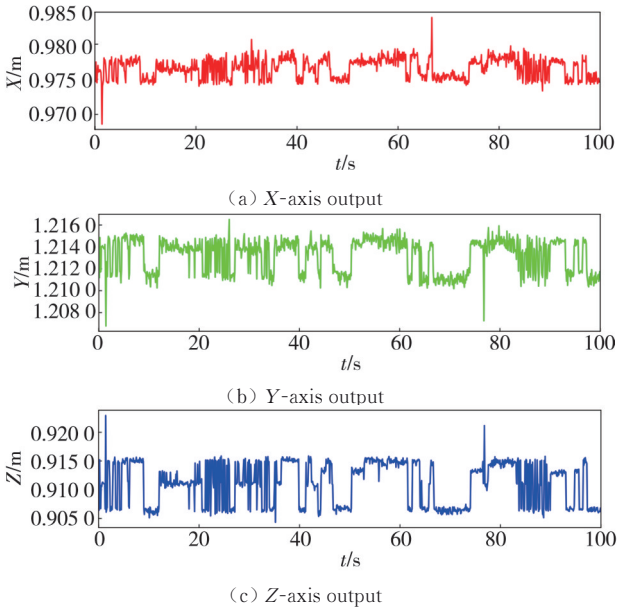


Fig. 10 Three-axis coordinate outputs in interference condition

The stability of the data was similarly evaluated using RMSE, and the results are shown in Table 2. It was observed that the algorithm using three information planes for pose estimation achieved higher data stability than the results obtained by using only two information planes. However, it should be noted that the enhanced initialization algorithm computed the pose with two information planes only under extreme angles and lighting conditions. So, under normal conditions, its stability is the same as that of the basic algorithm.

Table 2 RMSE in interference condition

Axis	RMSE/cm
X	0.143 00
Y	0.153 75
Z	0.369 44

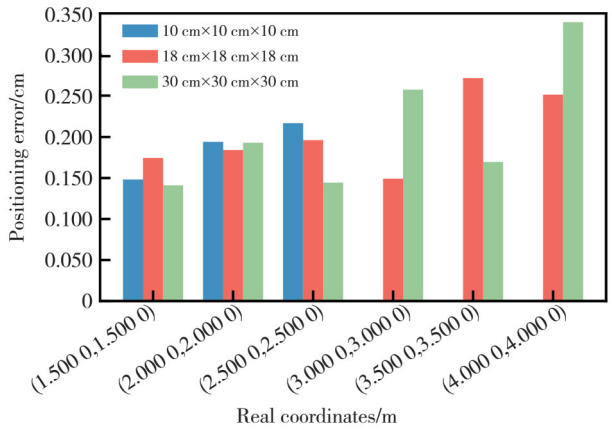
To verify the accuracy of the obtained pose at multiple sampling points, we selected eight sampling points with known coordinates, and obtained 50 consecutive frames to calculate the mean values of the measured coordinates, using the basic algorithm. The comparison between the measured results and the true coordinates, as well as the error values represented by Euclidean distance, are shown in Table 3. The results showed that the camera pose estimation algorithm based on the 3D target could achieve accurate localization.

Table 3 Measurement errors of positions

Real coordinates/m	Measured coordinates/m	Error/cm
(1.300 0, 1.700 0, 1.000 0)	(1.300 60, 1.699 30, 0.999 2)	0.122
(1.250 0, 1.650 0, 1.000 0)	(1.250 20, 1.649 90, 0.999 0)	0.102
(1.200 0, 1.600 0, 1.000 0)	(1.199 50, 1.600 60, 1.000 2)	0.081
(1.250 0, 1.600 0, 1.000 0)	(1.250 00, 1.599 60, 1.000 7)	0.081
(1.300 0, 1.600 0, 1.000 0)	(1.299 90, 1.598 50, 1.001 7)	0.227
(1.350 0, 1.600 0, 1.000 0)	(1.350 00, 1.598 40, 1.001 0)	0.189
(1.400 0, 1.600 0, 1.000 0)	(1.397 50, 1.598 70, 1.002 8)	0.397
(1.350 0, 1.650 0, 1.000 0)	(1.348 60, 1.648 50, 1.001 4)	0.248

To further verify the accuracy of 3D AprilTag detection, different-sized 3D targets were fabricated, with dimensions of 10 cm×10 cm×10 cm, 18 cm×18 cm×18 cm, and 30 cm×30 cm×30 cm, and were used to calculate the camera pose at a fixed coordinate system. For each experiment, a set of known fixed points at varying distances from the target was employed, covering a broad range of distances. By comparing the pose estimation errors for each point, the impact of target size and distance on positioning accuracy, as well as the relationship between target size and effective detection range, was assessed. The experimental results are shown in Fig.11, which demonstrate that the size of the target affects the maximum detectable distance.

Larger targets provide a larger effective detection range, but excessively large sizes may reduce the accuracy of feature point detection under the same manufacturing process. Therefore, the size of the 3D target must be adjusted based on practical requirements. The experiments also revealed that, within the effective detection range, positioning errors remained within a relatively stable range. Although the error tended to increase with distance, it stayed within an acceptable threshold. This indicated that, at an appropriate size, the target ensured both practical detection range and high recognition accuracy. Accordingly, for the subsequent dynamic trajectory experiments, the 18 cm×18 cm×18 cm target size was chosen, as it allowed for reliable detection within a 4 m range. In practical applications, adjusting the camera's position to ensure that the target pattern appears in an optimal area of the camera's view to achieve more accurate initialization is not technically difficult. This detection range specification satisfies operational requirements.



Note: The 10 cm×10 cm×10 cm target could not be detected at coordinates (3.000 0,3.000 0) and beyond; therefore, positioning errors were not calculable.

Fig. 11 Positioning error comparison for different target sizes

3.4 Construction of experimental scenarios and data description

The accuracy verification experiments for visual positioning systems typically utilize public datasets. These datasets contain continuous-time image frame sequences required by visual positioning systems, and each image frame is annotated with the true camera coordinates and pose information, which are used for accuracy evaluation.

However, the experimental design in this paper did not use public datasets. This is because the proposed initialization method based on the 3D target focuses on achieving efficient localization with scale information by simplifying the architecture of devices, which in turn eliminates the complexity of multi-sensor fusion. Existing public datasets do not include 3D targets, making it difficult to verify the characteristics and advantages of the proposed method.

We designed an experimental environment based on actual industrial scenarios to ensure that the experimental data reflect the complexity and challenges of real-world applications, thereby validating the applicability and reliability of the method. Additionally, given that the core of the proposed method lies in simplifying the design of devices, and its goal is to verify whether it can achieve accuracy comparable to that provided by multi-sensor systems in complex environments, it is not necessary to make direct horizontal comparisons with traditional multi-sensor fusion solutions.

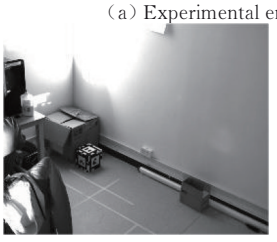
To overcome these limitations, we adopted a more flexible experimental design: the camera was mounted on a height-adjustable telescopic metal rod, allowing for control over the vertical positioning. The rod was securely attached to a wooden base equipped with four omnidirectional wheels, forming a stable mobile platform. This design ensured that the camera's height remained constant during its movement along a predefined trajectory.

The movement of the platform was manually controlled, with careful attention paid to maintaining a consistent speed. The motion was guided by guiding lines and coordinate points on the floor, which ensured that the camera followed its intended path with minimal deviation. This configuration was specifically chosen for its ability to provide controlled and repeatable motion, which was crucial for testing the initialization method under controlled experimental conditions. The guide lines and coordinates were measured using a graduated metal ruler, ensuring high precision. The resulting standard measurements had an accuracy of ± 0.5 mm, enhancing the reliability of the trajectory data.

This setup effectively simulated real-world scenarios requiring precise camera positioning, with the design balancing simplicity and effectiveness. By minimizing external factors such as camera displacement and ensuring consistent motion, we were able to conduct experiments that reliably tested the proposed initialization algorithm. The straightforward nature of the setup helped avoid unnecessary complexity while ensuring precision and repeatability.

Experiments were conducted in a semi-structured laboratory containing a large number of experimental apparatuses, ensuring that the test environment featured both the lighting stability of structured environments and the significant variability in object positions and shapes characteristic of unstructured environments. This setup made the experimental environment comparably challenging to commonly used indoor datasets, thereby aiding in the verification of the proposed method's generalization ability.

(a) Experimental environment overview



(b) Image captured at point *a*

(c) Image captured at point *b*

Fig. 12 Experimental environment and sample images

As shown in Fig.12, the experimental environment and the trajectory are presented. The images actually captured by the camera at the positions marked with numbers are shown as examples, in which the 3D target and various apparatuses can be observed.

Through the above experimental design, the proposed algorithm's performance in complex environments can be validated, being able to provide strong evidence for the method's generality and reliability.

3.5 Dynamic trajectory tracking experiment

To evaluate the positioning accuracy of the SLAM system with the proposed method during motion, a dynamic trajectory tracking experiment was conducted.

The camera captured images at a frame rate of 10 frames per second. To simplify data processing, the relative height of the camera was kept constant during the experiment. This setup simulated the horizontal movement of a device wearer in an indoor environment, ensuring consistency of experimental conditions and comparability of the data. The experimental results are shown in Fig.13. The positioning accuracy was evaluated by calculating the root mean square absolute trajectory error (RMS ATE)[21] between the world point coordinate sequence of the localization results and the reference trajectory of the predetermined path. The RMS ATE was 0.574 8 cm.

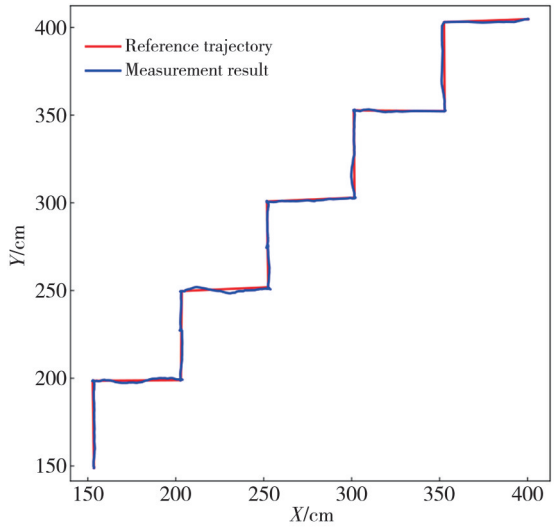


Fig. 13 Trajectory tracking results of ORB-SLAM3

The experimental results showed that the proposed method can provide real-scale information and achieve centimeter localization accuracy while maintaining real-time performance.

3.6 Cross-framework evaluation

In theory, the initialization method based on 3D target can be integrated into different SLAM frameworks, providing them with real-scale information. In previous sections of this chapter, the feasibility of the algorithm has been validated using ORB-SLAM3. To further evaluate its applicability, we integrated it into semi-direct visual odometry (SVO) and verified it under the same experimental environment. SVO is another mainstream visual odometry system, as a lightweight front-end framework primarily designed for camera pose estimation and feature matching. It lacks back-end global optimization capabilities; therefore, its localization performance is relatively weak because of the accumulated error of the front end.

The trajectory of the SVO system after integrating the proposed algorithm is shown in Fig. 14, with an RMS ATE of 8.73 cm. Notably, in the V102 sequence of the EuRoC dataset, the RMS ATE of SVO was 14 times that of ORB-SLAM3[19]. This indicates that the significant difference in RMS ATE observed in this experiment is due to the fact that ORB-SLAM3 has backend optimization, while SVO, as a lightweight framework, lacks it. This limitation is inherent to the framework itself and does not reflect the performance of the proposed initialization method.

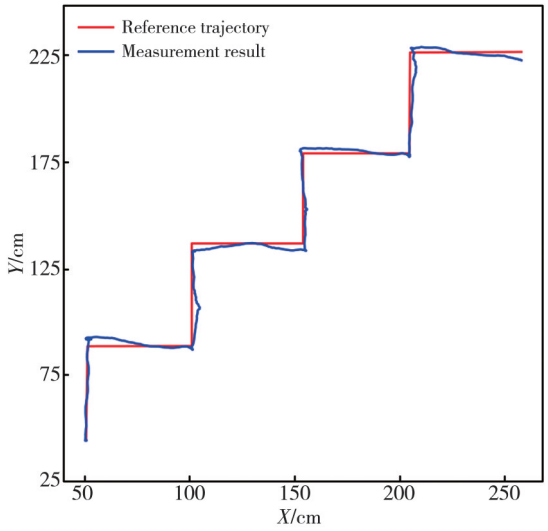


Fig. 14 Trajectory tracking results of SVO

The experimental results showed that after integrating the proposed algorithm, SVO could utilize the 3D target to achieve real-scale recovery. Although its accuracy was lower than that of ORB-SLAM3, which had back-end optimization capabilities, this experiment validated the applicability of the proposed algorithm to different types of visual SLAM frameworks.

Our study focused on visual positioning, and the integration of information from IMU with visual positioning information required a complex calibration process and high-precision sensor support, as a result, to reasonably evaluate

the differences in localization performance between visual-inertial odometry (VIO) systems and the proposed algorithm, we referred to publicly available experimental results from relevant literature for indirect comparison. For tabular comparison consistency, all RMS ATE values in Table 4 were standardized in meters.

As quantified in Table 4, mainstream visual-inertial systems achieve centimeter-level accuracy on the EuRoC dataset. In comparison, the proposed algorithm directly utilized the geometric properties of the 3D target to recover real-scale information and achieved comparable localization accuracy in real-world environments.

Table 4 RMS ATE across visual localization frameworks

Framework	RMS ATE/m	
	Public dataset (EuRoC V102)	Self-collected dataset
VI-DSO[22]	0.067	—
VINS-Mono[23]	0.066	—
ORBSLAM-VI[24]	0.028	—
ORB-SLAM3+Proposed method	—	0.006
SVO+Proposed method	—	0.087

The above experimental results showed that monocular SLAM frameworks integrated with the proposed algorithm could achieve absolute localization in real-world environments with accuracy close to that of VIO systems. Furthermore, the proposed method showed higher robustness during initialization by directly utilizing the scale information provided by the 3D target. Compared to VIO systems, it could avoid initialization failures caused by inertial noise accumulation. These results further validated the applicability of the proposed method.

4 Conclusions

To address the inherent limitations of traditional monocular visual measurement systems, namely complex initialization and the inability to recover real-scale information, this paper proposes an initialization method based on a 3D target. By designing a cubic 3D target, the system initialization and scale recovery could be efficiently completed using only two target images. The proposed initialization method based on the 3D target demonstrated framework-agnostic modularity enabling seamless integration into diverse SLAM architectures. As long as a 3D target is deployed in the target scene, real-scale recovery can be achieved in different scenarios without relying on complex multi-sensor fusion or high-performance computing devices. Experimental results showed that the monocular real-time localization system based on the 3D target accurately recovered real-scale information, achieving centimeter-level localization accuracy. Compared to multi-sensor fusion positioning systems, the proposed method exhibited significant advantages in terms of equipment cost and calibration simplicity. This method is expected to broaden the practical application scope of monocular visual localization systems and to provide theoretical support and practical value for the development of high-precision indoor positioning technology.

Acknowledgement

This work was supported by the Science and Technology Program Project of Tianjin (No. 24ZXZSSS00300).

Declaration of conflicting interests

The authors have no conflict of interests related to this publication.

References

[1] LI X X, GE M R, DAI X L, et al. Accuracy and reliability of multi-GNSS real-time precise positioning: GPS, GLONASS, BeiDou, and Galileo. Journal of Geodesy, 2015, 89(6): 607-635.

[2] MENG Q, LI S Y, JIANG Y Y, et al. Smartphone-based GNSS/PDR integration navigation enhanced by measurements resilient adjustment under challenging scenarios. GPS Solutions, 2024, 29(1): 23.

[3] OU C W, CHAO C J, CHANG F S, et al. A ZigBee position technique for indoor localization based on proximity learning// 2017 IEEE International Conference on Mechatronics and Automation, August 6-9, 2017, Takamatsu, Japan. New York: IEEE, 2017: 875-880.

[4] BELLAVISTA-PARENT V, TORRES-SOSPEDRA J, PEREZ-NAVARRO A. New trends in indoor positioning based on WiFi and machine learning: a systematic review// 2021 International Conference on Indoor Positioning and Indoor Navigation, November 29 - December 2, 2021, Lloret de Mar, Spain. New York: IEEE, 2022: 1-8.

[5] CASTILLO-CARA M, LOVÓN-MELGAREJO J, BRAVO-ROCCA G, et al. An empirical study of the transmission power setting for bluetooth-based indoor localization mechanisms. Sensors, 2017, 17(6): 1318.

[6] PLANGGER J, GOVINDASAMY RAVICHANDRAN H, RODIN S C, et al. System design and performance analysis of indoor real-time localization using UWB infrastructure// 2023 IEEE International Systems Conference, April 17-20, 2023, Vancouver, BC, Canada. New York: IEEE, 2023: 1-8.

[7] GUO Y G, LI Z C, ZHAO W B, et al. Two-laser-tracker system for precise coordinates transmission. Optics and Precision Engineering, 2020, 28(1): 30-38.

[8] SAPUTRA M R U, MARKHAM A, TRIGONI N. Visual SLAM and structure from motion in dynamic environments: a

survey. ACM Computing Surveys, 2019, 51(2): 1-36.
[9] SCHÖNBERGER J L, FRAHM J M. Structure-from-motion revisited//2016 IEEE Conference on Computer Vision and Pattern Recognition, June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE, 2016: 4104-4113.
[10] AL-TAWIL B, HEMPEL T, ABDELRAHMAN A, et al. A review of visual SLAM for robotics: evolution, properties, and future applications. Frontiers in Robotics and AI, 2024, 11: 1347985.
[11] LU X Y, WANG H, TANG S M, et al. DM-SLAM: monocular SLAM in dynamic environments. Applied Sciences, 2020, 10(12): 4252.
[12] NAWAL M, BRAHMANAGE G, LEUNG H. RGB-PD SLAM: scale consistent monocular SLAM using predicted depth//2024 International Conference on Consumer Electronics, July 9-11, 2024, Taichung, China. New York: IEEE, 2024: 359-360.
[13] LI R H, WANG S, LONG Z Q, et al. UnDeepVO: monocular visual odometry through unsupervised deep learning//2018 IEEE International Conference on Robotics and Automation, May 21-25, 2018, Brisbane, QLD, Australia. New York: IEEE, 2018: 7286-7291.
[14] KIM U H, KIM S H, KIM J H. SimVODIS: simultaneous visual odometry, object detection, and instance segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(1): 428-441.
[15] KENDALL A, CIPOLLA R. Geometric loss functions for camera pose regression with deep learning//2017 IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017, Honolulu, HI, USA. New York: IEEE, 2017: 6555-6564.
[16] ARANDJELOVIĆ R, GRONAT P, TORII A, et al. NetVLAD: CNN architecture for weakly supervised place recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(6): 1437-1451.
[17] TIAN R, ZHANG Y Z, ZHU D L, et al. Accurate and robust scale recovery for monocular visual odometry based on plane geometry//2021 IEEE International Conference on Robotics and Automation, May 30 - June 5, 2021, Xi′an, China. New York: IEEE, 2021: 5296-5302.
[18] LIU T, KUANG J, GE W F, et al. A simple positioning system for large-scale indoor patrol inspection using foot-mounted INS, QR code control points, and smartphone. IEEE Sensors Journal, 2021, 21(4): 4938-4948.
[19] CAMPOS C, ELVIRA R, RODRÍGUEZ J J G, et al. ORB-SLAM3: an accurate open-source library for visual, visual–inertial, and multimap SLAM. IEEE Transactions on Robotics, 2021, 37(6): 1874-1890.
[20] OLSON E. AprilTag: a robust and flexible visual fiducial system//2011 IEEE International Conference on Robotics and Automation, May 9-13, 2011, Shanghai, China. New York: IEEE, 2011: 3400-3407.
[21] STURM J, ENGELHARD N, ENDRES F, et al. A benchmark for the evaluation of RGB-D SLAM systems//2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 7-12, 2012, Vilamoura-Algarve, Portugal. New York: IEEE, 2012: 573-580.
[22] VON STUMBERG L, USENKO V, CREMERS D. Direct sparse visual-inertial odometry using dynamic marginalization//2018 IEEE International Conference on Robotics and Automation, May 21-25, 2018, Brisbane, QLD, Australia. New York: IEEE, 2018: 2510-2517.
[23] QIN T, LI P L, SHEN S J. VINS-mono: a robust and versatile monocular visual-inertial state estimator. IEEE Transactions on Robotics, 2018, 34(4): 1004-1020.
[24] MUR-ARTAL R, TARDÓS J D. Visual-inertial monocular SLAM with map reuse. IEEE Robotics and Automation Letters, 2017, 2(2): 796-803.

单目视觉定位的增强:基于立体靶标的初始化方法

马珺禹, 孙长库, 王 鹏*, 付鲁华

天津大学 精密测试技术及仪器全国重点实验室, 天津 300072

摘 要: 单目视觉定位系统因其设备成本低、标定简单而受到广泛关注。然而,传统单目系统因缺乏真实尺度信息和初始化流程复杂,在需要绝对定位的场景中应用受到限制。现有的解决方案存在两方面的权衡:不依赖靶标的方法依赖固定相机高度等环境先验约束实现尺度恢复,而基于靶标的尺度恢复方法通常要求靶标图案全程处于相机视野内。为应对这些挑战,提出了一种基于立体靶标的初始化方法,通过两帧靶标图像即可实现真实尺度的恢复。这种模块化方法可集成至多种单目同步定位与建图(Simultaneous location and mapping, SLAM)框架。将该方法与ORB-SLAM3以及半直接法视觉里程计(Semi-direct visual odometry, SVO)集成,开展验证工作,实验结果表明,所提方法在保证系统实时性的前提下提供真实尺度信息,适用于室内导航、工业机器人定位等对绝对定位有需求的应用场景。

关键词: 单目同步定位与建图;实时定位;尺度恢复;立体靶标;绝对定位;视觉里程计

引用格式: MA Junyu, SUN Changku, WANG Peng, et al. Enhancing monocular visual positioning: 3D target-assisted initialization. Journal of Measurement Science and Instrumentation, 2026, 17(1): 36-48. DOI: 10.62756/jmsi.1674-8042.2026003