STEREO VISUAL ODOMETRY BASED ON DYNAMIC AND STATIC FEATURES DIVISION

HUI XU, GUANGBIN CAI*, XIAOGANG YANG
ERLIANG YAO AND XIAOFENG LI

College of Missile Engineering, Rocket Force University of Engineering
Xi’an, Shaanxi 710025, China

(Communicated by Bin Li)

Abstract. Accurate camera pose estimation in dynamic scenes is an important challenge for visual simultaneous localization and mapping, and it is critical to reduce the effects of moving objects on pose estimation. To tackle this problem, a robust visual odometry approach in dynamic scenes is proposed, which can precisely distinguish between dynamic and static features. The key to the proposed method is combining the scene flow and the static features relative spatial distance invariance principle. Moreover, a new threshold is proposed to distinguish dynamic features. Then the dynamic features are eliminated after matching with the virtual map points. In addition, a new similarity calculation function is proposed to improve the performance of loop-closure detection. Finally, the camera pose is optimized after obtaining a closed loop.

Experiments have been conducted on TUM datasets and actual scenes, which shows that the proposed method reduces tracking errors significantly and estimates the camera pose precisely in dynamic scenes.

1. Introduction. Visual simultaneous localization and mapping (VSLAM) has received increased attention recently in the field of robotic autonomous navigation [8]-[36]. Visual odometry (VO) and loop-closure detection (LCD) are important components of VSLAM. VO estimates the pose of the robot from consecutive image frames [32], [30], and LCD detects whether the robot returns to the previous areas [22], [3], reducing the cumulative errors caused by the motion and improving the estimation accuracy of VO. Most state-of-the-art VSLAM methods assume static environments [8], [10]-[13]. However, there are many kinds of moving objects, including people and cars. Features on the moving objects will cause large pose estimation errors when faced with dynamic scenes. In addition, the appearance or disappearance of scene information can easily result in LCD failure. Therefore, the applications of existing VSLAM methods are limited. When the camera moves, the foreground and background in the camera’s field of view will move; hence, effectively distinguishing between dynamic and static features becomes a crucial issue of VO in dynamic scenes. In general, the small proportion of moving objects is viewed as a type of noise. To overcome this noise, an optimization method that uses laser data

2020 Mathematics Subject Classification. Primary: 65D19, 65D18; Secondary: 68U10.
Key words and phrases. Visual odometry, loop-closure detection, dynamic scenes, scene flow, static features relative spatial distance invariance principle, virtual map points.
The first author is mainly supported by NSSF of China under Grant (No. 61773387).
* Corresponding author: Guangbin Cai.
to calibrate the camera was proposed in [2]. However, the use of new sensors is costly and needs to deal with the convex constraints of the nonlinear equation [11]. There are similar processing methods for high-order information in [15]-[17]. Besides, the random sample consensus (RANSAC) method is usually applied [9], [27]. For example, the prior adaptive RANSAC (PA-RANSAC) algorithm was proposed by Tan [35], but the typical application of this method in real-life is not good enough owing to the constraints on the dynamic scenes in experiments. The RANSAC regression scheme and the iterative-closest-point (ICP) algorithm were combined to obtain the pose estimation for dynamic scenes in [19]. However, this method cannot filter out information regarding moving objects efficiently or calculate the camera pose precisely. In [20], Deok-Hwa proposed a VO algorithm using an inertial measurement unit to reduce the influence of moving objects. This approach also had some problems in that a new sensor was required and pose estimation was less accurate. There are some other methods that can handle the moving objects. Deok-Hwa proposed a background model-based dense visual odometry (BaMVO) to obtain robust navigation in dynamic scenes [21]. A nonparametric model was used to obtain the background subtraction to eliminate the moving objects. Then, the energy-based dense visual odometry (DVO) method [18] was applied to estimate camera pose. A novel static point weighting VO (SPW-VO) method based on matched depth edge points [5], [7] was proposed to downweight dynamic points on the moving objects [23]. Depth edge points were present sparsely and an efficient LCD procedure was performed in [21], resulting in a fast and robust SLAM system for dynamic scenes. A motion removal VO approach (MR-VO) was proposed in [34] based on dense moving-object segmentation for dynamic environments. However, when the dynamic features outnumber static features and the static environment assumption is totally violated, the previously described approaches cannot perform well for pose estimation. The SLAM system in dynamic environments (SLAMIDE) was proposed by Bibby [4] and takes advantage of expectation maximization (EM) and estimates the motion state of features in the scene robustly. This method establishes a probability distribution model for features, which can classify the features more accurately. Alcantarilla [1] introduced scene flow into VO to detect dynamic features by comparing the changes in scene flow. However, when there are few static features that can be used for matching, this method results in large errors, causing pose estimation to fail. A VO method combining scene flow and virtual map points (SF-VO) was proposed by Lin [26]. First, the scene flow was introduced to construct a Gaussian mixture model (GMM). Then the virtual map points were constructed, and the dynamic features were matched with the virtual map points. It was also included in the optimization framework of pose estimation. Peng used static point three-dimensional (3D) spatial relative distance invariance principle (SPDIP) to obtain static features and to eliminate the influence of dynamic features [31]. However, compared with the method of Lin [26], the degree of accuracy is low.

Based on the method of Lin [26] and combined with the work of Peng [31], in this article we propose an improved stereo VO algorithm and introduce an improved LCD algorithm to reduce the errors of pose estimation further. First, the scene flow of the matching features is calculated, and the preliminary classification of the feature points is realized according to the angle of the scene flow. Second, the
threshold of the dynamic and static features is calculated by SPDIP, and the features are matched with the virtual map points, which contributes to determining the motion states of the features more accurately. Then, the key-frames, (in which the dynamic feature points are distinguished), expressed by the bag of words (BOW) method, are included in the historical sets. Moreover, the similarity calculation function is improved to calculate the similarity between the current and historical frames, and the LCD is performed. Finally, the matched features are added to the pose optimization framework, and accurate camera poses are estimated by minimizing the re-projection error and the local bundle adjustment method. Through the experiments on datasets and lab scenes, the proposed method obtains the camera pose robustly in dynamic scenes.

The main contributions of this article are summarized as follows. (1) Combining the SPDIP and the scene flow allows us to distinguish accurately between the dynamic and static features. (2) We propose a new threshold for the division of motion points. (3) We apply a new similarity calculation function for LCD.

2. Preliminaries.

2.1. Stereo camera model. A stereo camera, as shown in Figure 1, is generally composed of two cameras placed horizontally. Stereo camera include binocular camera and RGBD camera. These two cameras have similar internal principles for obtaining depth information. The algorithm in this paper mainly uses the depth information of the image. If the stereo camera is corrected, that is, $O_l$ and $O_r$, the optical centers of the two cameras, are located on the X-axis. The distance between the two optical centres is called the baseline and is denoted as $b$.

![Figure 1. Stereo camera model](image)

Define the coordinate system of the left camera as the camera coordinate system. Define the coordinate system when the left camera obtains the first frame image as the world coordinate system. The spatial points $P$ obtained by the two cameras are denoted by $p_l(u_l, v)$ and $p_r(u_r, v)$, respectively. The projections of $P$ on the two cameras have some differences only in the abscissa. Here $d = u_l - u_r$ represents the parallax. The depth $z_c$ of the spatial points $P$ can be calculated as follows:

$$z_c = fb / d.$$  

(1)

In addition, there is a relation between the 3D point and the pixel coordinate as follows:

$$\pi (u_l, v, z_c) = \begin{pmatrix} \frac{(u_l - cx) \cdot z_c}{f} \\ \frac{(v - cy) \cdot z_c}{f} \\ z_c \end{pmatrix},$$

(2)
where \( f \) is the focal length, and \( c_x \) and \( c_y \) refer to the optical centers. Similarly, the 3D point \( P_c(x_c, y_c, z_c) \) in the camera coordinate system can be projected into the image plane to obtain \( (u, v) \) by the projection function, which is defined as follows:

\[
\pi^{-1}_i(P_c) = \left( \frac{f \cdot x_c}{z_c} + c_x, \frac{f \cdot y_c}{z_c} + c_y \right).
\] (3)

2.2. LCD algorithm based on BOW. The conventional BOW algorithm is trained with a large number of pictures, extracts the features of the images, and clusters them to obtain visual words, thus forming a visual dictionary. Then, the BOW method uses the statistical histogram of the words to describe the scenes when a new keyframe arrives. LCD is performed by calculating the similarity between the query frame and the historical frames.

The number of the tree branches is denoted by \( k \). The number of layers is denoted by \( L \). The \( k \) means clustering algorithm is called recursively for each branch, thereby \( k \) sub-branches are obtained with the \( L \) layers, as shown in Figure 2. The cluster center of each branch feature is taken as the node of the branch, and the description vector is used as a visual word to establish a visual dictionary tree.

The \( d \) dimension features are extracted from the image \( X \), where \( X = \{ x_1, x_2, \cdots, x_n \} \), \( x_i \in \mathbb{R}^d \). A visual dictionary tree is constructed according to the method as shown in Figure 2. The term frequency inverse document frequency (TF-IDF) entropy of the image at each tree node is taken as the score weight of the image in the visual word, which is defined as follows:

\[
\omega^l_i(X) = \frac{n_i}{n} \log \frac{N}{N_i},
\] (4)

where \( l \in \{0, 1, \cdots, L\} \), \( i \in \{1, 2, \cdots, k^l\} \), \( \omega^l_i(X) \) represents the projection score of the image \( X \) at \( O^l_i \), which denotes the \( i \)th node in the \( l \)th layer of the visual dictionary tree. Here \( N \) represents the total number of images to be processed, \( n_i \) represents the number of features projected by the image, and \( N_i \) represents the number of images with at least one feature projected onto the \( i \)th node.

The scene description vector of the image is as follows:

\[
W(X) = (W^1(X), W^2(X), \cdots, W^L(X)),
\] (5)

where \( W^l(X) = (\omega^1(X), \omega^2(X), \cdots, \omega^l(X)) \) represents the score vector of the image in the \( l \)th layer of the visual dictionary tree.

By calculating the similarity between the visual word score vectors of the two frames, LCD is performed to determine whether a closed loop occurs. However, there are many similarities in the actual scenes, causing perceptual ambiguity. This results in a false-positive closed loop, which will eventually lead to destruction of the map consistency.
According to [24], the similarity score at a single node of the image $X$ and $Y$ can be obtained by the minimum function as follows:

$$S_l^i(X,Y) = \min \{ \omega_l^i(X), \omega_l^i(Y) \}.$$  \hspace{1cm} (6)

The irrationality of this method was analyzed in [25], and an inverse proportional function was used to replace the similarity score function:

$$S_l^i(X,Y) = \frac{1}{|\omega_l^i(X) - \omega_l^i(Y)| + 1}.$$  \hspace{1cm} (7)

However, when the scenes are similar, that is, $\omega_l^i(Y_1) > \omega_l^i(Y_2) > \omega_l^i(X)$, and the difference between $|\omega_l^i(X) - \omega_l^i(Y_1)|$ and $|\omega_l^i(X) - \omega_l^i(Y_2)|$ is small, the difference between $S_l^i(X,Y_1)$ and $S_l^i(X,Y_2)$ is almost negligible. Therefore, the similarity score obtained by Equation (7) is low, which cannot solve the perceptual ambiguity caused by such high-level similar scenes. A new similarity calculation function is proposed in Section 3.5 to solve such problems.

3. Stereo VO in dynamic scenes. As shown in Figure 3, the proposed algorithm is mainly divided into three parts: the dynamic and static points discrimination, the improved LCD, and camera pose estimation considering virtual points. After introducing the scene flow and SPDIP to detect the features on the dynamic objects, the motion points can be matched with the virtual map points, and the static features are used to construct the local map to establish the matches between static feature points and the local map. The algorithm also uses static information to describe the scene for LCD, and after the closed-loop confirmation, the pose and map points are optimized.

![Figure 3. Overview of the proposed algorithm in dynamic scenes](image)

3.1. Preliminary detection of moving objects based on the scene flow. For any matching feature pairs $(u', v')$ and $(u, v)$ between two adjacent frames at time $k - 1$ and $k$, $P_{k-1}(x_{k-1}, y_{k-1}, z_{k-1})$ and $P(x, y, z)$, the coordinates of the camera coordinate system at the corresponding time can be obtained by Equations
Here $P_{k-1}(x_{k-1}, y_{k-1}, z_{k-1})$ is converted into the current moment camera coordinate system as follows:

$$P'(x', y', z') = R_{k-1}^k P_{k-1}(x_{k-1}, y_{k-1}, z_{k-1}) + t_{k-1}^k,$$  \(8\)

where $R_{k-1}^k$ denotes the rotation matrix from the moment $k - 1$ to the moment $k$ and $t_{k-1}^k$ denotes the translation vector between the two moments.

Put $P(x, y, z)$ and $P'(x', y', z')$ in the point set $A$ and $B$, respectively, that is, $A = \{P_1, \cdots, P_m\}$, $A \in R^3$ and $B = \{P'_1, \cdots, P'_m\}$, $B \in R^3$. Then, the scene flow is calculated as follows:

$$W = \{w : w = P - P', P \in A, P' \in B\},$$  \(9\)

that is, for any matching point pair of adjacent frames, their scene flow is:

$$w = P - P' = \begin{pmatrix} x - x' \\ y - y' \\ z - z' \end{pmatrix}.$$  \(10\)

Owing to the matching error, the projection error, and other factors (such as camera correction error), the scene flow of the static object features is not strictly zero. Therefore, static and dynamic objects features cannot be distinguished accurately by the scene flow alone.

Assume that the distribution of scene flow (belonging to the features of the same moving object) obeys the Gaussian model (GM). The scene flow distribution of multiple dynamic objects in the same scene is subject to the GMM.

The angles of scene flow represent the direction of the motion vector, which tend to remain consistent, and are subject to the GM. Therefore, when there are multiple moving objects in the scene, the scene flow angles of different moving objects obey the GMM. The means and variances of the GMM can be solved by the EM algorithm. In this way, the feature points corresponding to the scene flow can be divided into different types. However, the vectors with different modulus may have the same angle. Therefore, the angle of the scene flow cannot be simply used to determine whether a feature belongs to a static or moving object.

![Figure 4](image-url)

**Figure 4.** Classification of the scene flow based on angles [26]

The scene flow modulo values of the matching features indicate the corresponding spatial point distance. The smaller the distance, the higher the probability that the features belong to a static object. The larger the distance, the higher the probability that the features belong to a moving object. Therefore, by the method of weighted
averaging, a threshold that distinguishes roughly between static points or dynamic points is obtained.

3.2. **Static point relative space distance invariance principle (SPDIP).** Camera motion in 3D space is a Euclidean transformation, and the distance between any two points in the scene is independent of the camera motion [31]. This means that the relative distance between any two static points is constant. Therefore, by detecting whether the spatial distance of the two feature points changes, it can be determined that the two feature points move. If one feature point belongs to a static object and the other belongs to a moving object, the distance changes as they move.

![Figure 5. Invariance of the relative spatial distance of the static points](image)

As shown in Figure 5, it is assumed that the camera moves to the right, and the moving object moves to the lower left. It can also be found that the distance between \( P_1 \) and \( P_2 \) does not change. However, the distance between \( P_1 \) and \( P_3 \), \( P_2 \) and \( P_3 \) change greatly. Thereby the motion point and the static point can be distinguished, that is, \( P_1, P_2 \) are the static features corresponding to the spatial points, and \( P_3 \) is the motion point. Finally, only the static matching features are used for the motion estimation.

3.3. **Construction of virtual map points.** To make full use of the motion detection of the previous frame, virtual map points construction is applied. Assume that the current time is the moment \( k \). By using the motion model of the moving object at \( k - 1 \) and the pose of camera at \( k \), the map points corresponding to the moving object at \( k + 1 \) is estimated, and this constitutes a virtual map point set.

Specifically, the map points at \( k - 1 \) and \( k \) are obtained by Equations (1), (2). Motion changes between two moments are obtained as follows:

\[
M_k = P_k - P_{k-1}. \tag{11}
\]

As the interval of adjacent frames is small, the dynamic objects move with an approximatively constant speed. With the motion change at \( k \), the motion change at \( k + 1 \) is estimated as follows:

\[
P_{k+1}^{(v)} = P_k + M_k. \tag{12}
\]

The construction of virtual map points is demonstrated as shown in Figure 6. As these map points are estimated with a motion model between time \( k \) and \( k + 1 \), they are called virtual map points.
3.4. Distinguish dynamic feature points. In the method of [26], the Gaussian values of static features are used as the threshold to distinguish static and dynamic features. The Gaussian values of the features in the moving object model are compared with the threshold. If the value is greater than the threshold, the features are classified as dynamic features. However, when distinguishing between the dynamic and static features, the method only considers the change of the scene flow angle and neglects the importance of the scene flow modulus values. Scene flow modulus values may be different even if their angles are the same. If a point with large modulus value is considered to belong to the static model by the scene flow angle, it is classified incorrectly. In the same way, static points are also mistaken for dynamic points.

In the method of [31], the relative distance between the spatial points is used to distinguish between the static and moving objects. However, owing to factors such as light changes or the camera correction, the position estimation of the static points has large errors. If the threshold is too large, some dynamic points will be considered as static points. Similarly, if the threshold is too small, some static point will be misclassified as dynamic points. It can be found that the key problem of the two methods is the selection of the threshold. Therefore, in this article we propose a new threshold selection method by combining the above two ideas, which distinguishes between static and dynamic points and efficiently reduces the misclassification of feature points.

First, the modulus values \( c \) and angles \( \theta \) of matching features scene flow are calculated. Second, the EM algorithm is used to solve the parameters of the GMM. Feature points of the current frame are classified with the scene flow angle, and the average of modulus values of each kind of feature are calculated. The type features with the smallest average value are considered as the most likely type of static points and are denoted as the set \( S = \{ p_n, n = 1, 2, 3, \ldots, k \} \), in which \( k \) is the number of feature points. The smallest average modulo value is used as a threshold \( \psi_1 \) for distinguishing between dynamic and static points initially. If the modulus values are smaller than \( \psi_1 \), the feature points are the static feature points; otherwise, they belong to the motion points candidate set and await further discrimination.

Next, the three static points are found by SPDIP in the set \( S \). The state of the feature points cannot be distinguished accurately by the modulus values or angles. Therefore, using the weighted average method, a new evaluation index \( H_n \) is constructed for the \( n \)th feature points in the set \( S \) as follows:

\[
H_n = \frac{c_n}{k} + \frac{|\theta_n|}{\sum_{i=1}^{k}|\theta_i|}, \quad n = 1, 2, 3, \ldots, k,
\]
where $c_n$ denotes the modulus value and $\theta_n$ denotes the angle of the scene flow corresponding to the $n$th feature points. Here $q_1, q_2, q_3$, selected by the three smallest values of $H_n$, are three static feature points. The spatial points corresponding to the three static feature points are the three static points. The two graphs in Figure 7 show the three static features found in the dataset experiments and lab scene experiments. The results show that the proposed evaluation indicators find the static points accurately.

For these three static points, the relative spatial distance changes between any two points are calculated. The variation of the three sets of relative distances is obtained. For further judging the state of the feature points, a new threshold $\psi_2$ is calculated as follows:

$$\psi_2 = \frac{1}{3} |(|Q_1 Q_2| - |Q_1' Q_2'|)| + \frac{1}{3} |(|Q_1 Q_3| - |Q_1' Q_3'|)| + \frac{1}{3} |(|Q_2 Q_3| - |Q_2' Q_3'|)|,$$

where $Q_1, Q_2, Q_3$ denote the spatial points corresponding to $q_1, q_2, q_3$ and $Q_1', Q_2', Q_3'$ denote the spatial points corresponding to $q'_1, q'_2, q'_3$, respectively. If the average distance between a point and the three static points is greater than $\psi_2$, the point is matched with the set of virtual map points for the further discrimination. If the matching succeeds, it is a moving point. Otherwise, it is a point with an unknown state.

The blue points in Figure 7 are the three static points found by the proposed method. Figure 8 shows the judgment results of the state of feature points. The green points represent the static feature points and the red points shows the dynamic feature points.

The proposed algorithm uses a set of keyframes that are co-viewed with the current frame, denoted by $K_{m1}$. When estimating the camera pose, the match of the current frame feature point set $T$ and the local map points set $I$ are considered. The match of the current frame feature point set and the virtual map point set are also optimized. Match projection error can be obtained as follows:

$$e_i = \pi^{-1}(R_c^w P_i + t_c^w) - p_i,$$
where $R^c_w$ denotes the rotation matrix converted from the world coordinate system into the camera coordinate system, $t^c_w$ denotes the translation vector converted from the world coordinate system into the camera coordinate system, $P^i$ represents the map point in the world coordinate system, and $p_i$ is the feature point in the image coordinate system.

By minimizing the sum of the projection errors of the matched pairs of points, a more accurate camera pose estimation approach can be obtained by the following equation:

$$ E = \min_{R,t} \sum_i \|e_i\| \Omega $$

$$ = \min_{R,t} \sum_i \|\pi^{-1}(R^c_w P^i + t^c_w) - p_i\| \Omega, \quad (16) $$

where $\Omega$ denotes the matrix that describes trust in the matching of reprojection errors with virtual map point sets. The greater the number of matching points of virtual map points in a certain area, the greater the likelihood that the area belongs to a moving object. The more likely the local map points match the current frame feature points, the more likely the mismatching error occurs. The image is divided into $10 \times 10$ blocks. In each image block $C_i$, the ratio $r_i$ denotes the proportion of matching points formed by the virtual map points accounting for all matching point pairs. Obviously, this should be inversely proportional between $\Omega_i$ and $r_i$. Then, the local map points are optimized and adjusted by local bundle adjustment in [29] for a more accurate camera pose.

### 3.5. Improved LCD algorithm

There are actually some feature points on the moving objects. Based on the method in Section 3.4, the dynamic feature points are removed. The TF-IDF entropy of each node in the visual dictionary tree is utilized after clustering. The TF-IDF entropy is used as the score weight of the image in the visual word, then the score vector is obtained to describe the scene.

In this article, we propose a negative exponential power function to calculate the similarity between two images as follows:

$$ S^i_l(X,Y) = e^{-|\omega^i_l(X) - \omega^i_l(Y)|}. \quad (17) $$

The proposed function has the same advantages as the inverse proportional function. On the one hand, the negative exponential power function makes the similarity...
scores of $X$ and $Y$ at the nodes $O_l^i$ inversely proportional to the difference, that is, the smaller the difference, the higher the similarity score.

On the other hand, the similarity score of a single node in two images is guaranteed to be controlled within the range $(0, 1]$. Moreover, the similarity score is prevented from being excessively affected by a certain node, which improves reliability of the image similarity score of two frames. In addition, to solve the perceptual ambiguity caused by a high degree of similar scenes, when $|\omega_l^i(X) - \omega_l^j(Y)|$ is close to zero, the slope of the proposed function is larger than the inverse proportional function. This means that the scores and differences are obvious, and the degree of discrimination is good, which is conducive to the elimination of perceived ambiguity.

4. Experiments and analysis. Experiments with several challenging scenes have been conducted on a desktop PC with a 2.5 GHz CPU and 4 GB of RAM. GPU acceleration is not used. The system is Ubuntu14.04. The proposed method is evaluated using the TUM dataset and actual lab scenes with a Bumblebee2 stereo camera. A comparison between ORB-SLAM2 and the proposed algorithm is provided, showing the effectiveness of the proposed algorithm.

![Figure 9. Experiment scene sets](image)

(a) Bumblebee2 stereo camera  
(b) Lab scene 1  
(c) Scenes with highly similarity  
(d) Lab scene 2

4.1. Evaluation of VO method. To evaluate the proposed VO method, the root mean square error (RMSE) of the relative pose error (RPE) metric is used. The proposed method was first compared with other methods using the TUM dataset, and then the effectiveness of the proposed method was investigated in actual lab scenes.
The TUM dataset [33] was produced by the Technical University of Munich, in which the fr3 series are typical dynamic scene sequences. It mainly includes sitting-static, sitting-xyz, sitting-rpy, sitting-halfsphere, walking-static, walking-xyz, walking-rpy, and walking-halfsphere, in which the motion of the person in the walking sequence is severe and in the sitting sequence is not.

### Table 1. Translation drift and rotational drift of VO method on TUM dataset

| Sequences          | RMSE of translational drift [m/s] | RMSE of rotational drift [°/s] |
|--------------------|-----------------------------------|--------------------------------|
|                    | DVO     | BaMVO  | SPW-VO | Our Method | DVO     | BaMVO  | SPW-VO | Our Method |
| sitting-static     | 0.0157  | 0.0248 | 0.0231 | 0.0112     | 0.6084  | 0.6977 | 0.7228 | 0.3356     |
| sitting-xyz        | 0.0453  | 0.0482 | 0.0219 | 0.0132     | 1.4980  | 1.3885 | 0.8466 | 0.5753     |
| sitting-rpy        | 0.1735  | 0.1872 | 0.0843 | 0.0280     | 6.0164  | 5.9834 | 5.6258 | 0.6811     |
| sitting-halfsphere | 0.1005  | 0.0589 | 0.0389 | 0.0151     | 4.6490  | 2.8804 | 1.8836 | 0.6103     |
| walking-static     | 0.3818  | 0.1339 | 0.0327 | 0.0293     | 6.3502  | 2.0833 | 0.8085 | 0.5500     |
| walking-xyz        | 0.4360  | 0.2326 | 0.0651 | 0.1034     | 7.6669  | 4.3911 | 1.6442 | 2.3273     |
| walking-rpy        | 0.4038  | 0.3584 | 0.2252 | 0.2143     | 7.0662  | 6.3898 | 5.6902 | 3.9555     |
| walking-halfsphere | 0.2628  | 0.1738 | 0.0527 | 0.1061     | 5.2179  | 4.2863 | 2.4048 | 2.2983     |

The actual scene is set as follows. The Bumblebee stereo camera has a fixed baseline of approximately 12 cm. A person moves forward, backward, left, and...
Figures 7(b) and 8(b) show that the proposed method detects the dynamic points accurately. Figures 10 and 11 show the experimental results based on ORB-VO and the proposed method, respectively.

As shown in Figures 10(a) and 11(a), the errors between the estimated camera poses and the ground truth are clear. The red lines indicate the errors. The black point is the real position of the camera. The blue line is the estimated camera position. Figures 10(b) and 11(b) show the estimated errors of the camera trajectory. A comparison of Figures 10(a) and 11(a) shows that the camera pose estimated by the proposed method is more concentrated. As shown in Figures 10(b) and 11(b), the estimated errors of ORB-VO are much larger than the proposed method. The estimated errors of the proposed method are a maximum of 0.09 m and mostly concentrated in the range of 0.01 m and less. In addition, the estimation RMSE of ORB-VO is 0.137827 m; however, the RMSE value of the proposed method is 0.006584 m. In other words, the accuracy of pose estimation is improved by almost a factor of 20.

In the analysis and comparison of the experimental results of the two kinds of actual scenes, it is easy to find that the proposed method overcomes the interference in the camera pose estimation caused by the moving objects. It can still maintain strong robustness and high accuracy even though there are moving objects in the scenes. The validity and the feasibility of the proposed method are fully proved.

4.2. LCD algorithm in scenes with a high degree similarity. The actual scene experiment was performed by a Bumblebee stereo camera. The corridor was selected as the scene with a high degree of similarity and the motion track of the camera is shown as the red line in Figure 9(c).

Figures 12 and 13 show the LCD results obtained by the two methods, respectively. The comparison between Figures 12(a) and 13(a) shows that two experiments are able to detect the occurrence of a closed loop. Through the comparison between (b) and (c) in Figures 12 and 13, it is easy to find that the frame obtained by proposed method is closer to the closed-loop query frame. In a scene with a high
Figure 12. Loop-closure detection result of the inverse proportional function

Figure 13. Loop-closure detection result of the negative exponential power function

degree of similarity, the similarity score obtained by the proposed function is more distinctive, allowing more accurate detection of the loop-closure keyframe.

To verify the effectiveness of the proposed method in a dynamic scene, the experiment is set as follows: the camera moves around the lab scene as shown in Figure 9(c) and there is a moving person in the scene. During the experiment, ORB-SLAM2 fails to detect the loop-closure owing to the interference of the moving person. The improved LCD algorithm excludes the interference of dynamic feature points by detecting the dynamic feature points.

As Figure 14 shows, the trajectories are obtained by the proposed algorithm and ORB-SLAM2 in the experiment. The red rectangular part is the area including a loop-closure and walking person.

As shown in Figure 14(a), ORB-SLAM2 does not distinguish between the feature points on the moving object, resulting in large trajectory errors. Therefore, ORB-SLAM2 fails to detect the loop-closure in the red rectangular area. Figure
14(b) shows the camera trajectory obtained by the detection and deletion of dynamic points without LCD. Although the closed loop does not occur, the obtained trajectory is more accurate than the trajectory obtained by ORB-SLAM2.

![Trajectory comparison](image)

Figure 14. Loop-closure detection result of the negative exponential power function

Figure 14(c) shows the camera trajectory obtained by the proposed algorithm, including the LCD part. The algorithm overcomes the influence of moving people and detects the closed loop; therefore, the obtained trajectory is more accurate. Figure 14(d) shows the detail of the red rectangle in Figure 14(c).

Figure 14 shows that the proposed LCD algorithm can detect the closed loop smoothly in the dynamic scenes, which proves the effectiveness of the proposed algorithm.

4.3. Evaluation of VSLAM system. The proposed VO algorithm and ORB-VO were applied to the TUM fr3 series dataset. As shown in Figures 15 and 16, the camera trajectories estimated by the two methods are compared with the ground truth.
As shown in Figure 15, when estimating the camera trajectory in walking sequences, ORB-VO has large errors. This is because ORB-VO employed some dynamic feature points to estimate the camera pose. The proposed method is obviously superior to ORB-VO. It detects the dynamic feature points accurately, meaning more accurate static matching point pairs ensures the lower estimation errors.

Figure 16 shows the results obtained in the sitting sequences. Moving objects occupy a small proportion in the scene; therefore, the motion scenes in the sitting sequences have little effect on the ORB-VO and the proposed method.

The proposed algorithm improves the accuracy of pose estimation by recognizing the moving object. As Figures 15 and 16 show, the proposed method is robust and accurate in dynamic scenes.

![Comparisons between estimated trajectories and the ground truth in walking sequences](image)

The absolute trajectory error (ATE) represents the absolute error between the estimated camera trajectory and the ground truth. The two algorithms were employed in the TUM fr3 series dataset to obtain the RMSE of the ATE, as listed in Table 2.

Black bold data indicates a better result. In comparison with ORB-VO and the algorithm of [29] in sitting sequences, the proposed method obtains a smaller RMSE, which indicates that the proposed method still has good performance in low dynamic scenes. When the proposed method is employed in walking dynamic scenes, the accuracy improves significantly. In the walking-static and walking-halfsphere datasets, the accuracy of the proposed method is improved by an order of magnitude.
Figure 16. Comparisons between estimated trajectories and the
ground truth in sitting sequences

Table 2. RMSE of the ATE of camera pose estimation (m^{-1})

| Sequences           | ORB-SLAM2 | MR-SLAM | SPW-SLAM | SF-SLAM    | Our Method |
|---------------------|-----------|---------|----------|------------|------------|
| sitting-static      | 0.0082    |         |          | 0.0081     | 0.0073     |
| sitting-xyz         | 0.0094    | 0.0482  | 0.0397   | 0.0101     | 0.0090     |
| sitting-rpy         | 0.0197    |         |          | 0.0180     | 0.0162     |
| sitting-halfsphere  | 0.0211    | 0.0470  | 0.0432   | 0.0239     | 0.0164     |
| walking-static      | 0.1028    | 0.0656  | 0.0261   | 0.0120     | 0.0168     |
| walking-xyz         | 0.4278    | 0.0932  | 0.0601   | 0.2251     | 0.0884     |
| walking-rpy         | 0.7407    | 0.1333  | 0.1791   | 0.1961     | 0.3620     |
| walking-halfsphere  | 0.4939    | 0.1252  | 0.0489   | 0.0423     | 0.0411     |

compared with ORB-VO. In the walking-xyz dataset, the accuracy is also improved by a factor of three compared with the method of Lin [26]. The experiment fully demonstrates that the proposed algorithm has better robustness and accuracy in high-level dynamic scenes.

In the walking-rpy dataset experiment, the camera pose accuracy obtained by the proposed method was lower than the results given in [23], [34], [26], but it is still twice the accuracy of ORB-VO. The camera rotates at different angles violently in the walking-rpy, and the people in the scene are also moving quickly in a wide range, resulting in multiple interruptions of the ORB-VO. Although the relocation can be reluctantly retrieved, the error accumulation is large. When the proposed algorithm runs, the scene changes quickly and the moving object occupies a larger part of the
view, resulting in fewer matching points between adjacent frames. This means that few feature points can be utilized for classification by the scene flow, which leads to the wrong threshold. When estimating the camera pose, some dynamic feature points are introduced, resulting in unfaithful camera pose estimation.

In the experiments on the datasets, the average time of processing each frame of the proposed algorithm is about 35.23 ms, which meets the basic real-time processing requirements.

5. Conclusions. In this article, an approach that can distinguish accurately between static and dynamic feature points has been proposed to help the VO achieve good pose estimation performance in dynamic scenes. Combined with the improved LCD method, the pose estimation has been further optimized. The combination of scene flow and SPDIP has been used to more accurately obtain dynamic and static feature points. During the LCD process, only static feature information is used to describe the scenes, and a negative exponential power function has been proposed as the similarity calculation function, which was able to effectively overcome ambiguity problems in highly similar scenes. The experiments have shown that the proposed stereo VO method can accurately estimate the camera poses in the dynamic scenes compared with ORB-VO.

Acknowledgments. This study was co-supported by the National Natural Science Foundation (NNSF) of China under Grant (No. 61773387), and China Postdoctoral Fund under Grant (Nos. 2016M590971 and 2017T100770).

REFERENCES

[1] P. F. Alcantarilla, J. J. Yebes, J. Almazán et. al., On combining visual slam and dense scene flow to increase the robustness of localization and mapping in dynamic environments, 2012 IEEE International Conference on Robotics and Automation, Saint Paul, Minnesota, USA, IEEE, 2012.
[2] Y. An, B. Li, L. Wang et.al., Calibration of a 3D laser rangefinder and a camera based on optimization solution, J. Ind. Manag. Optim., 17 (2021), 427–445.
[3] A. Angeli, D. Filliat, S. Doncieux et. al., Fast and incremental method for loop-closure detection using bags of visual words, IEEE Transactions on Robotics, 24 (2008), 1027–1037.
[4] C. Bibby and I. Reid, Simultaneous localisation and mapping in dynamic environments (SLAMIDE) with reversible data association, Robotics: Science and Systems, Atlanta, Georgia, USA, 2007.
[5] L. Bose and A. Richards, Fast Depth Edge Detection and Edge Based Rgb-D Slam, IEEE International Conference on Robotics and Automation, Stockholm, Sweden, IEEE, 2016.
[6] C. Cadena, L. Carlone, H. Carrillo et. al., Simultaneous localization and mapping: Present, future, and the robust-perception age, IEEE Transactions on Robotics, 32 (2016), 1309–1332.
[7] C. Choi, A. J. Trevor and H. I. Christensen, Rgbd Edge Detection and Edge-Based Registration, 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, IEEE, 2013.
[8] A. J. Davison, I. D. Reid, N. D. Molton et. al., MonoSLAM: Real-time single camera SLAM, IEEE Transactions on Pattern Analysis & Machine Intelligence, 29 (2007), 1052–1067.
[9] J. Engel, V. Koltun and D. Cremers, Direct sparse odometry, IEEE Transactions on Pattern Analysis & Machine Intelligence, 40 (2018), 611–625.
[10] J. Engel, T. Schöps and D. Cremers, LSD-SLAM: Large-Scale Direct Monocular SLAM, European Conference on Computer Vision, Springer, Zürich, Switzerland, 2014.
[11] J. Fan, On the Levenberg-Marquardt methods for convex constrained nonlinear equations, J. Ind. Manag. Optim., 9 (2013), 227–241.
[12] C. Forster, M. Pizzoli and D Scaramuzza, SVO: Fast Semi-Direct Monocular Visual Odometry, IEEE International Conference on Robotics and Automation, Hong Kong, China, IEEE, 2014.
[13] C. Forster, Z. Zhang, M. Gassner et. al., SVO: Semi-direct visual odometry for monocular and multicamera systems, IEEE Transactions on Robotics, 33 (2017), 249–265.

[14] J. Fuentes-Pacheco, J. Ruiz-Ascencio and J. M. Rendón-Mancha, Visual simultaneous localization and mapping: A survey, Artificial Intelligence Review, 43 (2015), 55–81.

[15] D.-K. Gu, G.-P. Liu and G.-R. Duan, Robust stability of uncertain second-order linear time-varying systems, J. Franklin Inst., 356 (2019), 9881–9906.

[16] D.-K. Gu and D.-W. Zhang, Parametric control to second-order linear time-varying systems based on dynamic compensator and multi-objective optimization, Appl. Math. Comput., 365 (2020), 124681, 25 pp.

[17] M. Labbe and F. Michaud, Appearance-based loop closure detection for online large-scale and long-term operation, IEEE Transactions on Robotics, 9 (2013), 734–745.

[18] S. Li and D. Lee, RGB-D slam in dynamic environments using static point weighting, IEEE Robotics and Automation Letters, 2 (2017), 2263–2270.

[19] B. Li, D. Yang and L. Deng, Visual vocabulary tree with pyramid TF-IDF scoring match scheme for loop closure detection, Acta Automatica Sinica, 37 (2011), 665–673.

[20] Z. L. Lin, G. L. Zhang, E. Yao et. al., Stereo visual odometry based on motion object detection in the dynamic scene, Acta Optica Sinica, 37 (2017), 187–195.

[21] M. Lourakis and X. Zabulis, Model-Based Pose Estimation for Rigid Objects, International conference on computer vision systems, St. Petersburg, Russia, Springer, 2013.

[22] R. Mur-Artal, J. M. M. Montiel and J. D. Tardós, ORB-SLAM: A versatile and accurate monocular slam system, IEEE Transactions on Robotics, 31 (2015), 1147–1163.

[23] R. Mur-Artal and J. D. Tardós, ORB-SLAM2: An opensource slam system for monocular, stereo, and rgbd cameras, IEEE Transactions on Robotics, 33 (2017), 1255–1262.

[24] D. Scaramuzza and F. Fraundorfer, Visual odometry, IEEE Robotics & Automation Magazine, 18 (2011), 80–92.

[25] J. Sturm, N. Engelhard, F. Endres et. al., A Benchmark for the Evaluation of RGB-D SLAM Systems, IEEE International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, IEEE, 2012.

[26] Y. Sun, M. Liu and M. Q. H. Meng, Improving rgbd slam in dynamic environments: A motion removal approach, Robotics and Autonomous Systems, 89 (2017), 110–122.

[27] W. Tan, H. Liu, Z. Dong et. al., Robust Monocular SLAM in Dynamic Environments, IEEE International Symposium on Mixed and Augmented Reality, Adelaide, Australia, IEEE, 2013.

[28] G. Younes, D. Asmar, E. Shammas et. al., Keyframe-based monocular slam: Design, survey, and future directions, Robotics and Autonomous Systems, 98 (2017), 67–88.