Robotics and Autonomous Systems 107 (2018) 209–220

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Robust RGB-D visual odometry based on edges and points

Erliang Yao [a,*], Hexin Zhang [a], Hui Xu [a], Haitao Song [a], Guoliang Zhang [b]

[a] Department of Control Engineering , High-Tech Institute of Xi'an, Xi'an, China
[b] College of Controlling Engineering , Chengdu University of Information Technology, Chengdu, China

HIGHLIGHTS

- We propose a novel reference frame selection scheme to handle the motion blur.
- An area state detection is presented as a sparse motion segmentation approach.
- The proposed method obtains precise localization in dynamic environments.

ARTICLE INFO

Article history:
Received 31 January 2018
Received in revised form 9 May 2018
Accepted 19 June 2018
Available online 26 June 2018

Keywords:
Localization
Visual odometry
Dynamic environments
Edge alignment
Bundle adjustment

ABSTRACT

Localization in unknown environments is a fundamental requirement for robots. Egomotion estimation based on visual information is a hot research topic. However, most visual odometry (VO) or visual Simultaneous Localization and Mapping (vSLAM) approaches assume static environments. To achieve robust and precise localization in dynamic environments, we propose a novel VO based on edges and points for RGB-D cameras. In contrast to dense motion segmentation, sparse edge alignment with distance transform (DT) errors is adopted to detect the states of image areas. Features in dynamic areas are ignored in egomotion estimation with reprojection errors. Meanwhile, static weights calculated by DT errors are added to pose estimation. Furthermore, local bundle adjustment is utilized to improve the consistencies of the local map and the camera localization. The proposed approach can be implemented in real time. Experiments are implemented on the challenging sequences of the TUM RGB-D dataset. The results demonstrate that the proposed robust VO achieves more accurate and more stable localization than the state-of-the-art robust VO or SLAM approaches in dynamic environments.

1. Introduction

In unknown or GPS-denied environments, accurate localization with visual data is one of the most active topics in Robotics. RGB-D cameras could directly provide depth information of environments, and have been used in more and more robotic applications. As a part of vSLAM (visual Simultaneous Localization and Mapping) [1], VO (Visual Odometry) [2] focuses on the egomotion estimation of consecutive images. To simplify the formulation, most SLAM and VO approaches assume static environments, namely, the information among consecutive images is consistent. However, moving objects exist everywhere, such as pedestrians, animals and cars. The image information factors that change at different times consist of the camera motion and moving objects. The moving objects destroy the static environment assumption and have negative effects on VO. Traditional VO methods obtain poor localization in dynamic environments.

Based on optimized errors, traditional VO or SLAM methods can be categorized into roughly four types. The first type is based on reprojection errors, namely, indirect methods, such as PTAM [3] and ORB-SLAM2 [4]. Point features of consecutive images are extracted and matched with invariant feature descriptors. Reprojection errors of point features are minimized for precise egomotion estimation. Due to robust point features, these methods allow large inter-frame movements, and the features can be used for loop detection to further improve the localization accuracy. The second type is based on photometric errors, namely, direct methods, e.g. DVO [5], LSD-SLAM [6] and DSO [7]. Direct methods apply intensity values of raw image data to estimate the camera motion without feature extraction. Compared with the reprojection location of features in indirect methods, direct methods adopt the intensity gradient direction and magnitude to guide the egomotion estimation. However, they assume small inter-frame motion and photometric invariance. In addition, effective loop closures cannot be integrated into direct methods. The third type is based on Euclidean space errors with the ICP (iterative closest point) algorithm, such as

* Corresponding author.
E-mail address: familyyao915@126.com (E. Yao).

https://doi.org/10.1016/j.robot.2018.06.009
0921-8890/

RGB-D SLAM [8] and KinectFusion [9]. Instead of aligning the image information as with the above two methods, this approach registers point clouds. Due to the depth uncertainty of the 3D features [10], the egomotion estimation is less accurate than that of minimizing reprojection errors. However, optimization with large sets of points and a 3D Gaussian error model could improve the performance of the egomotion estimation [11]. The fourth type of VO or SLAM method is based on distance transform (DT) errors, e.g., REVO (Robust Edge-based Visual Odometry) [12] and D-EA (Direct Edge Alignment) [13,14]. These methods are considered crossovers between direct and indirect methods [12]. They extract edges in images and calculate the distance transform [15], which provides the edge similarity measure without correspondence matching. The edges between a source frame and a target frame are aligned for egomotion estimation by minimizing DT errors. Compared with indirect methods, no matching steps are needed for the calculation of the DT errors, which saves the computation cost. In addition, REVO exhibits a larger convergence basis than direct and indirect methods when increasing inter-frame motion. However, due to the large number of 3D edge points, BA is hardly applied in edge alignment [14]. Besides, further effort is required to the loop closure in edge alignment.

In highly dynamic environments, the standard RANSAC (RANdom SAmple Consensus) [16] approach does not handle dynamic objects well [17–19]. The traditional robust kernel functions, such as the Huber robust kernel function, cannot eliminate the influence of highly dynamic objects, as shown in Section 4.1. To handle moving objects in dynamic environments, different strategies are adopted for robust VO or SLAM. Namdev et al. [20] segment moving objects based on dense optical flow and two view geometry. The optical flow potential and geometry potential are calculated for a graph based segmentation algorithm. Similar potentials are clustered together and motion segmentation is obtained. Their method achieves the precise segmentation of moving objects. However, a frame needs seven minutes to be computed. Sun et al. [21] calculate the difference image between a source image and a target image to detect the boundary of moving objects. Then, motion segmentation is achieved based on vector quantized depth images, but a frame costs almost half a second. Moreover, its performance in lowly dynamic scenes is not satisfactory. These two dense motion segmentation methods are time-consuming. RDSLAM [18] compares appearance and structure to detect the changed GPUSITF features. Outliers are removed by a prior-based adaptive RANSAC algorithm. However, RDSLAM is restricted to small environments and is not sufficiently accurate. Li et al. [19] extract foreground depth edges. Then, the static point weights of the depth edge points are calculated based on Student's t-distribution. The Static point weights, intensity weights and geometric weights are added to ICP for the registration. The experiments on the TUM RGB-D dataset [22] show that this method achieves perfect results. However, loop closure based on 3D points is more simplistic than the methods based on point features.

In contrast to previous robust approaches of egomotion estimation in dynamic environments, we propose a novel robust VO based on edges and points for RGB-D cameras. A two-stage process is adopted in this method as follows: edge alignment by DT errors detects dynamic areas in images, and camera poses are estimated with static weights by the reprojection errors of non-dynamic areas. Compared with dense motion segmentation, edge alignment dealing with sparse 3D points guarantees real-time performance. Besides, egomotion estimation with little static information in dynamic environments is a challenging task. Reprojection error optimization with a small amount of static and accurate point features can obtain better localization results than the other three types. The main contributions of this work are as follows:

(1) To handle motion blur in images, a novel reference frame selection strategy is proposed for the edge alignment in dynamic environments.

(2) As sparse motion segmentation, a novel area state detection scheme is presented based on DT errors and priors, and the states are divided further based on reprojection errors.

(3) Experiments on the TUM dataset show that the proposed method obtains more precise localization than other robust approaches.

To the best of our knowledge, the proposed robust VO is the first method combining edges and points in dynamic environments.

2. Preliminaries

The world coordinate system and camera coordinate system are defined by W and C respectively. A camera pose in the world coordinate system is defined as a transformation matrix $\boldsymbol{T}_{\mathrm{W}}^{\mathrm{C}} \in SE(3)$, consisting of an orthogonal rotation matrix $\boldsymbol{R}_{\mathrm{W}}^{\mathrm{C}} \in SO(3)$ and a translation vector $\boldsymbol{t}_{\mathrm{W}}^{\mathrm{C}}$:

$$\boldsymbol{T}_{\mathrm{W}}^{\mathrm{C}} = \begin{bmatrix} \boldsymbol{R}_{\mathrm{W}}^{\mathrm{C}} & \boldsymbol{t}_{\mathrm{W}}^{\mathrm{C}} \\ 0 & 1 \end{bmatrix} \tag{1}$$

For a 3D point $\boldsymbol{P}_{\mathrm{W}} = (x_{\mathrm{W}}, y_{\mathrm{W}}, z_{\mathrm{W}})$in the world coordinate system, the corresponding position $\boldsymbol{P}_{\mathrm{C}} = (x_{\mathrm{C}}, y_{\mathrm{C}}, z_{\mathrm{C}})$ in the camera coordinate system can be obtained by $\boldsymbol{T}_{\mathrm{W}}^{\mathrm{C}}$:

$$\boldsymbol{P}_{\mathrm{C}} = \boldsymbol{T}_{\mathrm{W}}^{\mathrm{C}} \circ \boldsymbol{P}_{\mathrm{W}} = \boldsymbol{R}_{\mathrm{W}}^{\mathrm{C}} \boldsymbol{P}_{\mathrm{C}} + \boldsymbol{t}_{\mathrm{W}}^{\mathrm{C}} \tag{2}$$

Accordingly, let us project $\boldsymbol{P}_{\mathrm{C}}$ into the image coordinate system and obtain pixel position $\boldsymbol{u} = (u, v)$ by $\boldsymbol{\pi}(\boldsymbol{P}_{\mathrm{C}})$:

$$\boldsymbol{u} = \boldsymbol{\pi}(\boldsymbol{P}_{\mathrm{C}}) = \left(\frac{x_{\mathrm{C}} f_x}{z_{\mathrm{C}}} + c_z, \frac{y_{\mathrm{C}} f_y}{z_{\mathrm{C}}} + c_y\right) \tag{3}$$

where f_x and f_y are focal lengths, and c_x and c_y are optical centres of a pinhole camera model. Conversely, if the depth z_{C} of a pixel $\boldsymbol{u}$ is known, $\boldsymbol{P}_{\mathrm{C}}$ can be calculated as follows:

$$\boldsymbol{P}_{\mathrm{C}} = \left(\frac{u - c_x}{f_x} \cdot z_{\mathrm{C}}, \frac{v - c_y}{f_y} \cdot z_{\mathrm{C}}, z_{\mathrm{C}}\right) \tag{4}$$

Ordinarily, VO based on optimization takes the egomotion estimation as an energy minimization problem, which needs a minimal representation of a camera pose. However, $\boldsymbol{T}_{\mathrm{W}}^{\mathrm{C}}$ is over-parameterized and could not be utilized in optimization. Therefore, Lie algebra is adopted to parameterize the camera pose. An element $\boldsymbol{\xi} \in \mathfrak{se}(3)$ of Lie-algebra can be mapped to $\boldsymbol{T} \in SE(3)$ by an exponential map. In contrast, $\boldsymbol{T}$ can be mapped to ξ by a logarithmic map:

$$\boldsymbol{T} = \exp_{\mathfrak{se}(3)}(\boldsymbol{\xi}) \tag{5}$$

$$\boldsymbol{\xi} = \log_{SE}(3)(\boldsymbol{T}) \tag{6}$$

A minimized function $F(\boldsymbol{T})$ could be solved with an iterative non-linear least square method. The element $\boldsymbol{\xi} \in \mathfrak{se}(3)$ can act as a perturbation of the transformation matrix $\boldsymbol{T}$. In each iteration, $\boldsymbol{\xi}$ is optimized to minimize the function energy, and then, $\boldsymbol{T}$ is updated with $\boldsymbol{\xi}$.

3. Robust visual odometry

3.1. Overview

ORB features [23] are selected for the representations of points. Fig. 1 demonstrates the overview of the proposed robust VO method. Similar to ORB-SLAM2, the VO approach consists of a tracking thread and a local mapping thread. First, ORB features and edge information of the current frame are extracted while DT of the edge information is calculated. Edge alignment based on DT

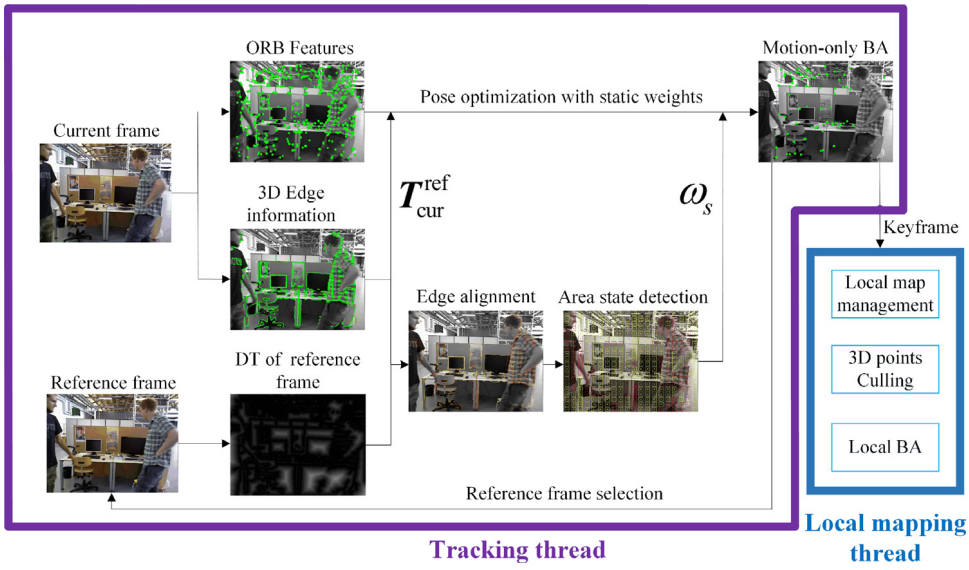


Fig. 1. Overview of robust VO method.

errors is executed between the current frame and a reference frame to acquire the transformation matrix $\boldsymbol{T}_{\text{cur}}^{\text{ref}}$. Static weights $\boldsymbol{\omega}_{\text{s}}$ are calculated by the DT errors and prior states from the last frame. The states of the current image areas are determined by $\boldsymbol{\omega}_{\text{s}}$. Then, the current camera pose is initialized with $\boldsymbol{T}_{\text{cur}}^{\text{ref}}$ and is optimized based on ORB features in non-dynamic areas with $\boldsymbol{\omega}_{\text{s}}$ by motion-only BA (bundle adjustment). Finally, if the current frame is chosen as a keyframe, it would be sent to the local mapping thread to adjust a local map and to execute local BA. Note that reference frames are different from keyframes as follows: reference frames are used for edge alignment, and keyframes participate in the local mapping.

3.2. Edge weight estimation

An image is divided into $20 * 20$ pixel blocks. For example, a pixel whose location is (10, 10) in an image at $640 * 480$ resolution belongs to the pixel block(1, 1). Each pixel block corresponds to an area of the image, which has the following three states: static, unknown and dynamic. Unknown areas mean that the areas are static or dynamic, which cannot be determined. The generation of unknown areas depends on the DT errors in Section 3.3, and the unknown areas are divided into static areas or dynamic areas by reprojected errors of the point features listed in Section 3.5.

As mentioned in [12], deep learned edge features, such as Structured Edges [24] and Holistically Nested Edges [25], omit weak edges in images. Their performances in the edge alignment are better than that of the Canny algorithm [26]. However, the lack of weak edges is unfit for the subsequent area state detection. Therefore, 2D edges in an image are still extracted by the Canny edge detector in this paper for each frame, as shown in Fig. 2(b). If a 2D edge has a reliable depth measure from a RGB-D camera, the corresponding 3D edge point is obtained by (4).

After edge extraction, its DT information based on 2D edges is calculated by the OpenCV library, as shown in Fig. 2(c). The intensity of each pixel in the DT image indicates the distance to the closest edge. The brighter the pixel in the DT image, the farther away from the closest edge. The intensity of the pixel serves as the measurement of the subsequent edge alignment, namely, the DT value.

A good edge distribution is of importance for edge alignment. However, when moving objects have many textures (such as the checked shirt in Fig. 2(c), the 3D edge points from the dynamic textures will introduce an uneven spatial distribution, which heavily affects edge alignment. Therefore, each pixel block contains N_{th} 3D edge points at most.

The transformation matrix $\boldsymbol{T}_{\text{cur}}^{\text{ref}}$ between a reference frame and current frame is initialized with the motion model of the camera. The motion model assumes that the camera motion is identical between the adjacent time intervals. Assuming that current time is k, we can obtain $\boldsymbol{T}_{\text{cur}}^{k-1} = \boldsymbol{T}_{k-1}^{k-2}$. Then, $\boldsymbol{T}_{\text{cur}}^{\text{ref}}$ is initialized as follows:

$$\boldsymbol{T}_{\text{cur}}^{\text{ref}} = \boldsymbol{T}_{\text{w}}^{\text{ref}}(\boldsymbol{T}_{\text{w}}^{k-1})^{-1}\boldsymbol{T}_{k-1}^{k-2} \tag{7}$$

$$\boldsymbol{T}_{k-1}^{k-2} = \boldsymbol{T}_{\text{w}}^{k-2}(\boldsymbol{T}_{\text{w}}^{k-1})^{-1} \tag{8}$$

Then, $\boldsymbol{T}_{\text{cur}}^{\text{ref}}$ is estimated by the edge alignment, which minimizes DT errors of the current 3D edge points $\boldsymbol{P}_{\text{C}}^{\text{cur}}$. For the current 3D edge point $\boldsymbol{P}_{\text{C}}^{\text{cur\_}i}$, its position in the coordinate system of the reference frame is obtained by $\boldsymbol{T}_{\text{cur}}^{\text{ref}} \circ \boldsymbol{P}_{\text{C}}^{\text{cur\_}i}$. Then it is projected into the reference frame to obtain the pixel coordinate $\boldsymbol{\pi}(\boldsymbol{T}_{\text{cur}}^{\text{ref}} \circ \boldsymbol{P}_{\text{C}}^{\text{cur\_}i})$, and its DT value is $DT_{\text{ref}}[\boldsymbol{\pi}(\boldsymbol{T}_{\text{cur}}^{\text{ref}} \circ \boldsymbol{P}_{\text{C}}^{\text{cur\_}i})]$. If $\boldsymbol{T}_{\text{cur}}^{\text{ref}}$ is precise, the reprojected pixel $\boldsymbol{\pi}(\boldsymbol{T}_{\text{cur}}^{\text{ref}} \circ \boldsymbol{P}_{\text{C}}^{\text{cur\_}i})$ of $\boldsymbol{P}_{\text{C}}^{\text{cur\_}i}$ should be an edge pixel, which means that $DT_{\text{ref}}[\boldsymbol{\pi}(\boldsymbol{T}_{\text{cur}}^{\text{ref}} \circ \boldsymbol{P}_{\text{C}}^{\text{cur\_}i})]$ is zero. Therefore, $DT_{\text{ref}}[\cdot]$ shows the errors introduced by $\boldsymbol{T}_{\text{cur}}^{\text{ref}}$, which is termed the DT error. The edge alignment is expressed as follows:

$$\boldsymbol{T}_{\text{cur}}^{\text{ref}} = \arg\min_{\boldsymbol{T}_{\text{cur}}^{\text{ref}}} \sum_{i} \omega_H^i \cdot \left\| DT_{\text{ref}}[\boldsymbol{\pi}(\boldsymbol{T}_{\text{cur}}^{\text{ref}} \circ \boldsymbol{P}_{\text{C}}^{\text{cur\_}i})] \right\|^2 \tag{9}$$

where ω_{H}^i is calculated by the Huber kernel function as follows:

$$\omega_{\text{H}}^i = \begin{cases} 1 & DT_{\text{ref}}[\cdot] \le \theta_{\text{th}} \\ \dfrac{\theta_{\text{th}}}{DT_{\text{ref}}[\cdot]} & DT_{\text{ref}}[\cdot] > \theta_{\text{th}} \end{cases} \tag{10}$$

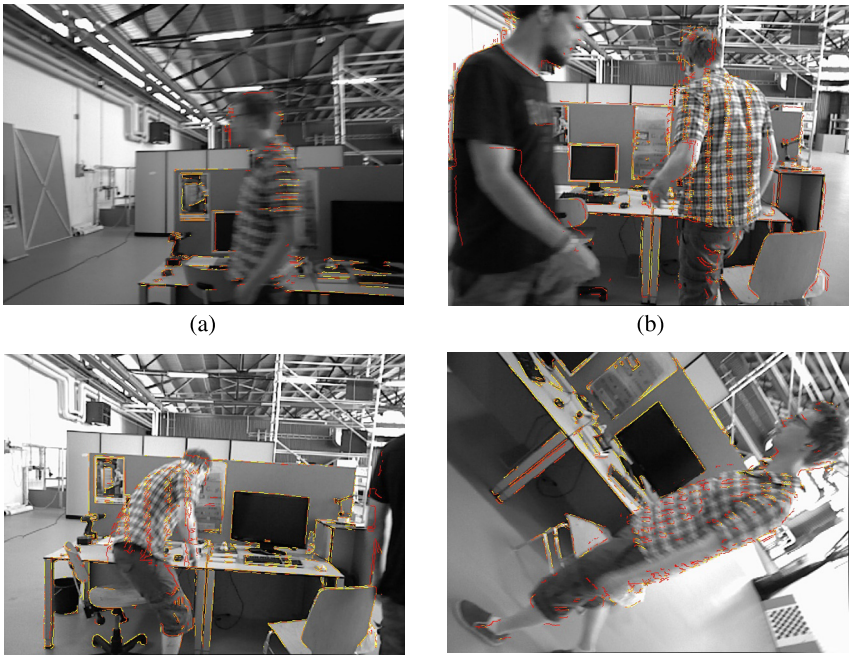
where θ_{th} is the threshold of the Huber kernel function.

Edge repeatability is important for edge alignment. Following REVO, an edge filter is utilized for enhancing reliability. However, compared with REVO taking all current 3D edge points into edge alignment, the proposed method only takes into account the current 3D edge points that are reprojected into the non-dynamic (namely, static and unknown) areas of the reference frame.

Edge alignment can be solved by an iterative non-linear optimization, such as Levenberg–Marquardt (LM) algorithm. Then, egomotion estimation for current frame$\boldsymbol{T}_{\text{cur}}^{\text{ref}}$ and a weight ω_{H}^i for each 3D edge point $\boldsymbol{P}_{\text{C}}^{\text{cur\_}i}$ are obtained. The reprojections of the current 3D edge points are shown in Fig. 3. Compared with static

(a) Gray image (b) 2D edge image (c) DT image

Fig. 2. Edge extraction and calculation of DT.



(a) (b) (c) (d)

Fig. 3. Current 3D edge points are projected into reference frames. The sequences are from TUM RGB-D dataset. The persons move in the environments. Red edges indicate high DT errors and yellow edges express low DT errors. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

edges, the edges of dynamic objects obtain higher DT errors. We can conclude that edge alignment based on DT errors works well in dynamic environments.

3.3. Area state detection

In this section, the areas will be divided into static areas, dynamic areas and unknown areas preliminarily. Here, an example is given to illustrate the area state detection, as shown in Fig. 4. Let us assume that there are a static area A, a dynamic area B and an unknown area C in the reference frame. For the current frame, there are extracted edges in the D, E and F areas. First, the 3D edge points in D and E are projected into A in the reference frame. Depending on the DT errors, D and E are divided into a static area and a dynamic area respectively. Second, we assume that 3D edge points in F are projected into B or C. Due to the uncertainty of dynamic objects, the states of current 3D edge points which are projected into B cannot be determined by the DT errors of the reference frame. Similarly, the state of C may be dynamic or static. Therefore, the state of F cannot be determined by B and C. Third, there are no edges in G, which means that DT errors cannot be calculated. Then, G is divided as an unknown area. As we can see, the unknown areas consist of the following two types: current areas without edges (namely, G), and current areas whose corresponding areas of the reference frame are non-static (namely, F).

First, we consider static areas and dynamic areas. For each pixel block (i, j) of the current frame, the weight and depth statistics of the 3D edge points are analysed to judge dynamic areas. The average weight $\overline{\omega}$ and average depth $\overline{d}$ of a pixel block (i, j) are computed as follows:

$$\overline{\omega}(i,j) = \sum_k \omega_{\mathrm{H}}^k / N_{i,j} \tag{11}$$

$$\overline{d}(i,j) = \sum_k d^k / N_{i,j} \tag{12}$$

where $N_{i,j}$ is the total number of effective 3D edge points in pixel block(i, j), which are projected into the static areas of the reference frame. d^k is the corresponding depth value of a 3D edge point in the current frame.

The DT errors of current edges are not enough to determine that a pixel block of the current frame is a dynamic area. Fig. 3 shows

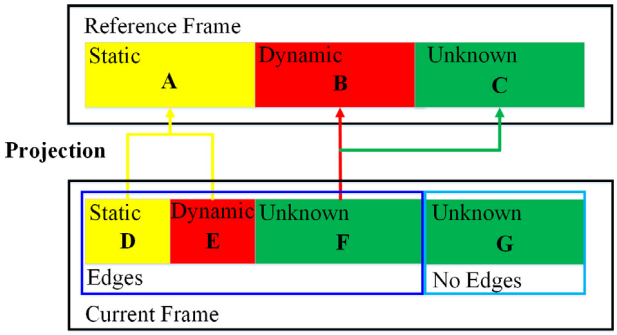


Fig. 4. Illustration of area state detection.

that, due to dynamic objects, edge alignment is not completely precise, which results in large DT errors of several static edges. Thus, the states of corresponding pixel blocks in the last frame are taken into account. The static weight and average depth of the last frame are $\omega_{\mathrm{ls}}(i,j)$ and $\overline{d}_{\mathrm{l}}(i,j)$, respectively. If the average depths $\overline{d}(i,j)$ and $\overline{d}_{\mathrm{l}}(i,j)$ are similar, the states of the current pixel block and the last pixel block should be consistent. Thus, the current static weight $\omega_{\mathrm{s}}(i,j)$of pixel block is calculated as follows:

$$\omega_{\mathrm{s}}(i,j) = (1 + \frac{\omega_{\mathrm{ls}}(i,j) - \omega_{\mathrm{th}}}{e^{d_{\mathrm{diff}}}}) \cdot \overline{\omega}(i,j) \tag{13}$$

$$d_{\mathrm{diff}} = \mathrm{abs}(\overline{d}(i,j) - \overline{d}_{\mathrm{l}}(i,j)) \tag{14}$$

where ω_{th} is a threshold to decide the state of a pixel block.abs$(\cdot)$ obtains the absolute value of a number. d_{diff} is the depth difference. The smaller the d_{diff}, the more consistent $\omega_{\mathrm{ls}}(i,j)$ and $\omega_{\mathrm{s}}(i,j)$ are. In contrast, if d_{diff} is large, $\omega_{\mathrm{ls}}(i,j)$ has little impact on $\omega_{\mathrm{s}}(i,j)$.

If $\omega_{\mathrm{s}}(i,j)$ is less than ω_{th}, the pixel block (i,j) is considered a dynamic area (namely, E in Fig. 4), as the red squares shown in Fig. 5. Otherwise, it is a candidate of static areas.

Then, unknown areas are analysed for non-dynamic areas. It is worthwhile to note that several dynamic areas are not detected, such as parts of the checked shirt in Fig. 5a and the arm in Fig. 5b. These areas are regarded as unknown areas in our method (namely, G in Fig. 4). These unknown areas result from non-existent edges. In fact, the subsequent pose optimization based on features depends on the non-dynamic areas. If there are no edges in a pixel block, features are nearly non-existent in this area. Therefore, the states of areas without edges hardly reduce the localization accuracy. Furthermore, if the projection of the current 3D edge point $\boldsymbol{P}_{\mathrm{C}}^{\mathrm{cur\_}i}$ belongs to the non-static areas of the reference frame, the state of $\boldsymbol{P}_{\mathrm{C}}^{\mathrm{cur_}i}$cannot be determined. If half of the 3D edge points in the current pixel block are unknown, the pixel block is considered an unknown area (namely, F in Fig. 4). The static weight $\omega_{\mathrm{s}}(i,j)$ of an unknown area is set as 0.5 to model the uncertainty for the pose optimization in Section 3.5. Candidates of static areas that are not categorized as unknown areas, are regarded as static areas (namely, D in Fig. 4).

3.4. Reference frame selection

In REVO [12], the tracking quality is measured after edge alignment between the current frame and a reference frame. If the quality is not valid, the last frame would be selected as a new reference frame. The DT of the new reference frame would be calculated and aligned with the current frame. Namely, double edge alignments and one calculation of DT are executed when the reference frame is updated. This strategy increases computation costs. Moreover, in dynamic environments, the poor egomotion estimation provided by edge alignment cannot accurately project 3D edge points for the tracking quality appraisal, which limits the evaluation of the strict overlaps of edges between the current frame and the reference frame. This results in frequent reference frame insertion.

A novel reference frame selection is proposed for edge alignment in dynamic environments, as shown in Fig. 6. Contrary to

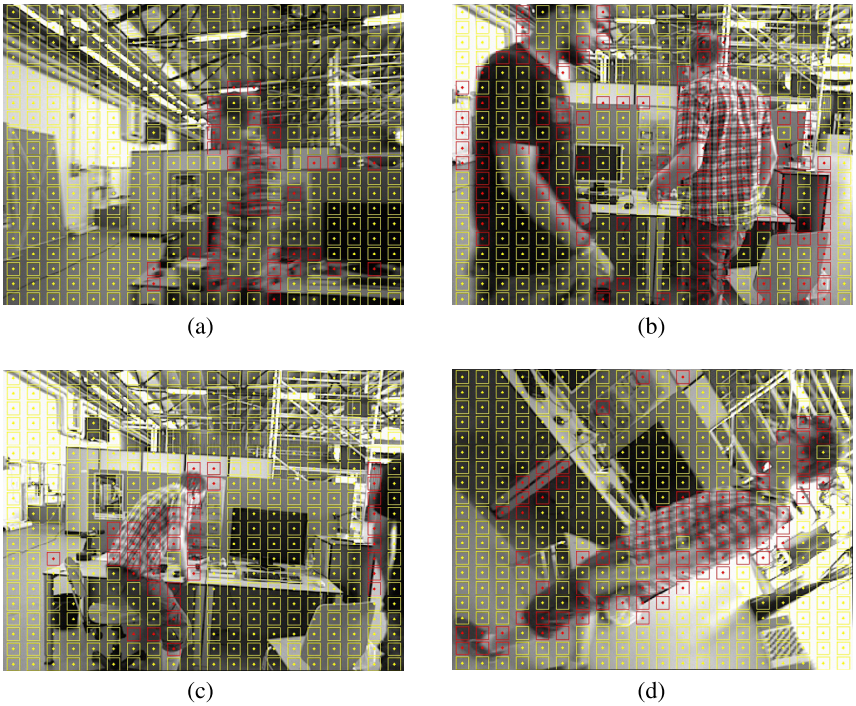


Fig. 5. Dynamic area detection based on static weights. Red squares indicate dynamic areas. Yellow squares imply static areas and unknown areas. To show the states of the areas clearly, each square does not cover the whole pixel block. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

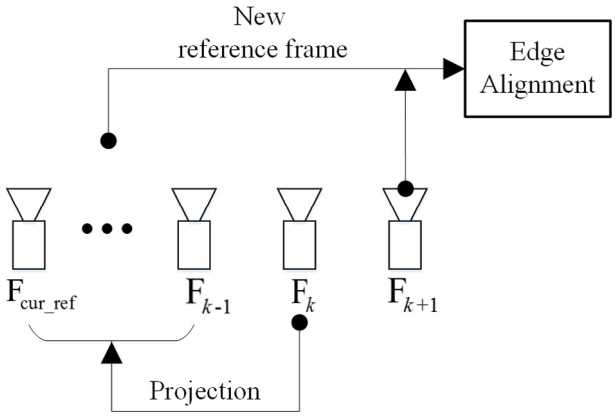


Fig. 6. Strategy of the proposed reference frame selection. Weighted DT errors among current frame F_k and candidate reference frames in $\mathbb{R} = \{F_{cur\_ref}, \ldots, F_{k-2}, F_{k-1}\}$ are evaluated. The frame with the least-weighted DT error is selected as a new reference frame for the next frame F_{k+1}.

REVO, a new reference frame is selected for edge alignment of the next frame. It is assumed that the image information between two successive frames changes little. Weighted DT errors are adopted for the matching evaluation between the current frame and adjacent frames. An adjacent frame with a least-weighted DT error is selected as the new reference frame.

In general, the weighted DT error between successive frames is the least. However, if motion blur exists in the last frame, edge alignment between the last frame and the current frame will become insufferable. As a result, area state detection based on DT errors is inaccurate. Therefore, a reference frame is selected from adjacent frames with the least-weighted DT error.

First, we assume that the current frame is the kth frame, termed F_k. Then, a set of candidate reference frames is obtained:

$$\mathbb{R} = \{F_{cur\_ref}, \ldots, F_{k-2}, F_{k-1}\} \tag{15}$$

Note that $\mathbb{R}$ does not consist of F_k, which means that edge alignment between two successive frames is not utilized by this method. The reason for this limit is that the dynamic areas can hardly be detected by edge alignment if the dynamic objects move slowly. However, when the camera stays still in static environments, $\mathbb{R}$ may contain many candidate reference frames, which results in high computation costs in reference frame selection. For this reason, $\mathbb{R}$ is restricted to a maximum of four candidate reference frames according to the adjacent time frames.

Second, the current 3D edge points in the non-dynamic areas are projected into candidate reference frames by transformations. DT errors are obtained and weights ω_H^i are calculated by (10). Then, the weighted DT error of the current frame and a candidate reference frame is evaluated by V_r that has an inverse correlation as follows:

$$V_r = \sum_i \omega_c^i \omega_H^i \tag{16}$$

where ω_H^i has an inverse correlation with the DT error. ω_c^i adjusts the importance of a 3D edge point based on the numerical interval of ω_H^i, as shown in (17). It is clear that, ω_c^i encourages large ω_H^i and restrains small ω_H^i. This means that a smaller DT error is more acceptable. The reason for this behaviour is that, the DT errors obtained by images with motion blur are generally large, as shown in the middle image of Fig. 7(a), and the proposed method prefers the reference frame with little or no motion blur.

$$\omega_c^i = \begin{cases} 1.5 & \omega_H^i \in [0.8, 1] \\ 1.25 & \omega_H^i \in [0.6, 0.8) \\ 1 & \omega_H^i \in [0.4, 0.6) \\ 0.5 & \omega_H^i \in [0, 0.4) \end{cases} \tag{17}$$

Finally, the candidate reference frame with the largest V_r is selected as the new reference frame. Fig. 7 shows an example of reference frame selection with motion blur. The frames are from the "Fr3/walking_static" sequence of the TUM dataset. As we can see in Fig. 7(a), the edge alignment by a reference frame (the 37th frame in the sequence) with motion blur is unsatisfactory. Many static areas are misjudged as dynamic areas while the reference frame (the 35th frame in the sequence) obtained by the proposed method is aligned properly.

3.5. Egomotion estimation based on point features with static weights

The tracking thread executes the motion-only BA for pose optimization, while the local mapping thread implements the local BA of camera poses and 3D feature points. Following ORB-SLAM2 [4], the motion-only BA consists of tracking the last frame and the local map. Contrary to ORB-SLAM2, states of image areas and static weights from edge alignment and area state detection are considered in optimization.

First, feature matching is conducted for non-dynamic areas by descriptors. Each matched feature whose location in an image is $\boldsymbol{u}_i$ associates a corresponding 3D feature point $\boldsymbol{P}_W^i$ from the last frame or keyframes.

Second, the reprojection errors of corresponding 3D feature points are minimized to estimate egomotion. If only ORB features in static pixel blocks take part in the pose estimation, the number of ORB features may sometimes not be sufficient for accurate optimization. Actually, the ORB features in both static and unknown pixel blocks participate in the optimization of the motion-only BA. Due to the uncertainties of ORB features in unknown pixel blocks, the ORB features in static pixel blocks should play a more important role in the egomotion estimation. To adjust the importance of ORB features in static and unknown pixel blocks, the static weight ω_s is adopted in the optimization.

$\boldsymbol{T}_W^{cur}$ is initialized as $\boldsymbol{T}_{ref}^{cur}\boldsymbol{T}_W^{ref}$ by $\boldsymbol{T}_{cur}^{ref}$ from the edge alignment. The motion-only BA is solved by the LM algorithm:

$$\boldsymbol{T}_W^{cur} = \arg\min_{\boldsymbol{T}_W^{cur}} \sum_i \omega_s(\mathrm{Grid}(\boldsymbol{u}_i)) \cdot \omega_{pyr} \cdot \left\| \boldsymbol{u}_i - \boldsymbol{\pi}(\boldsymbol{T}_W^{cur} \circ \boldsymbol{P}_W^i) \right\|^2 \tag{18}$$

where $\mathrm{Grid}(\cdot)$ obtains a pixel block position to which $\boldsymbol{u}_i$ belongs. ω_s determines the static weight of the pixel block. ω_{pyr} is the weight of an image pyramid according to ORB-SLAM2:

$$\omega_{pyr} = \frac{1}{s^{l_i}} \tag{19}$$

where s is the scale of an image pyramid and l_i is the level of the ORB feature in the image pyramid. The higher the level of the image pyramid, the more uncertain the ORB feature.

Third, after the pose optimization, a chi-square test for the reprojection errors is adopted to reject outliers. Outliers are divided into the following two types: outliers in static areas from mismatches and outliers in unknown areas from dynamic objects. The relationships between 3D feature points and features of the current frame are deleted. Moreover, for outliers of the second type, the 3D feature points do not participate in future pose optimization.

Lastly, the states of unknown areas are updated by the ratio of outliers in the pixel blocks. If half of the features in an unknown pixel block are outliers, the area is changed to a dynamic area.

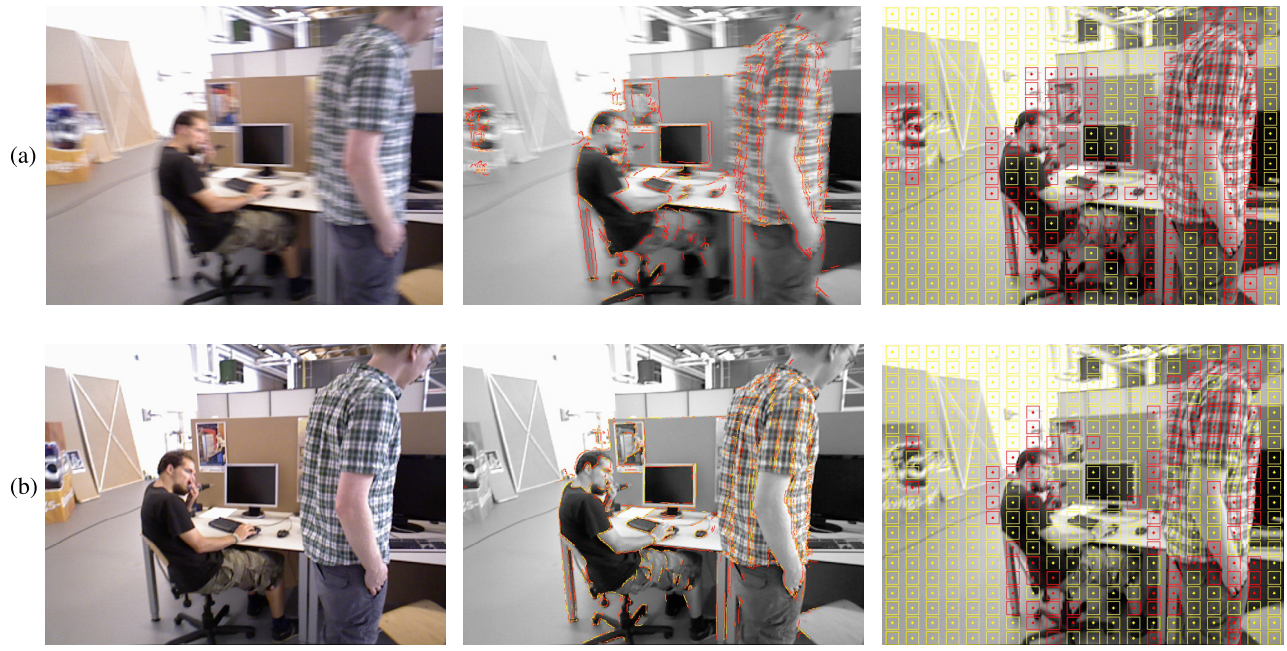


Fig. 7. Area state detection with different reference frame selection: (a) and (b) are obtained by REVO and our method respectively. The images in the first column are the selected reference frames. The images in the second column are the results of edge alignment. The images in the third column show the dynamic areas by red squares. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 8. Static ORB features for tracking local map in static areas. There are almost no features in dynamic objects, namely, the moving persons.

Otherwise, the unknown area become a static area. Fig. 8 shows the ORB features in static areas.

The local mapping thread is similar to that of ORB-SLAM2. However, only the features in static areas are used when creating new 3D feature points for triangulation. A local map is obtained by the keyframe covisibility information. Then, local BA is implemented to enhance the consistencies of keyframes and 3D feature points in the local map.

4. Experiment

The proposed robust VO method is tested on the TUM RGB-D benchmark, which is collected by a moving Kinect. Its time-synchronized ground truth is obtained by a motion capture system. It contains several challenging sequences for SLAM evaluation, such as large loop sequences, kidnapped sequences and sequences with dynamic objects. To validate the performance of the proposed VO, we principally focus on the static and dynamic sequences.

The dynamic sequences are categorized to several types by the movement styles of Kinect and dynamic objects:

1. Sitting: persons sitting on chairs pick and place objects, or talk to others.
2. Walking: persons move around the areas.
3. Xyz: The Kinect moves along the *x*-*y*-*z* axes.
4. Rpy: The Kinect rotates along the roll–pitch–yaw axes.
5. Halfsphere: The Kinect moves along halfsphere-like trajectories.
6. Static: The Kinect remains nearly static.

The TUM RGB-D dataset provides various dynamic environments, which can be applied to evaluate the accuracy and stability of the robust VO. In contrast, the traditional VO, robust VO and robust SLAM are tested by the same sequences.

4.1. Comparisons with traditional VO

Among traditional approaches, ORB-SLAM2 is a precise SLAM system based on indirect methods, which provides satisfactory results for static scenes of the TUM RGB-D dataset. It utilizes the Huber kernel function and the visibility of 3D feature points to reject outliers, which can handle lowly dynamic environments. The open-source implementation [27] is adopted for ORB-SLAM2.

The proposed method and ORB-SLAM2 are evaluated with several static scenes, a low-dynamic scene and a highly dynamic scene, as shown in Table 1. For a fair comparison, the loop closure detection of ORB-SLAM2 is closed in the experiments. The influences of dynamic objects increase gradually from static environments to dynamic environments. To quantify the localization accuracy, the Absolute Trajectory Error (ATE) [22] that represents

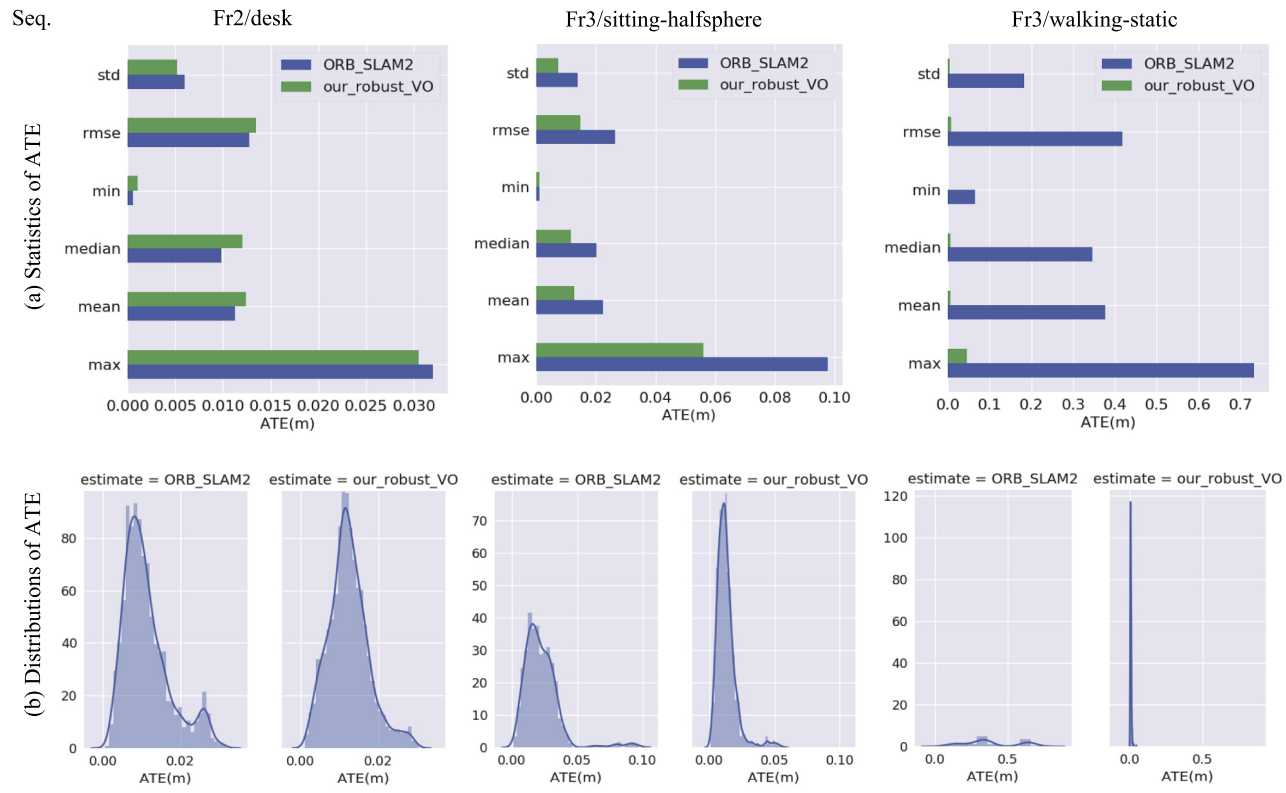


Fig. 9. The statistics and distributions of ATE.

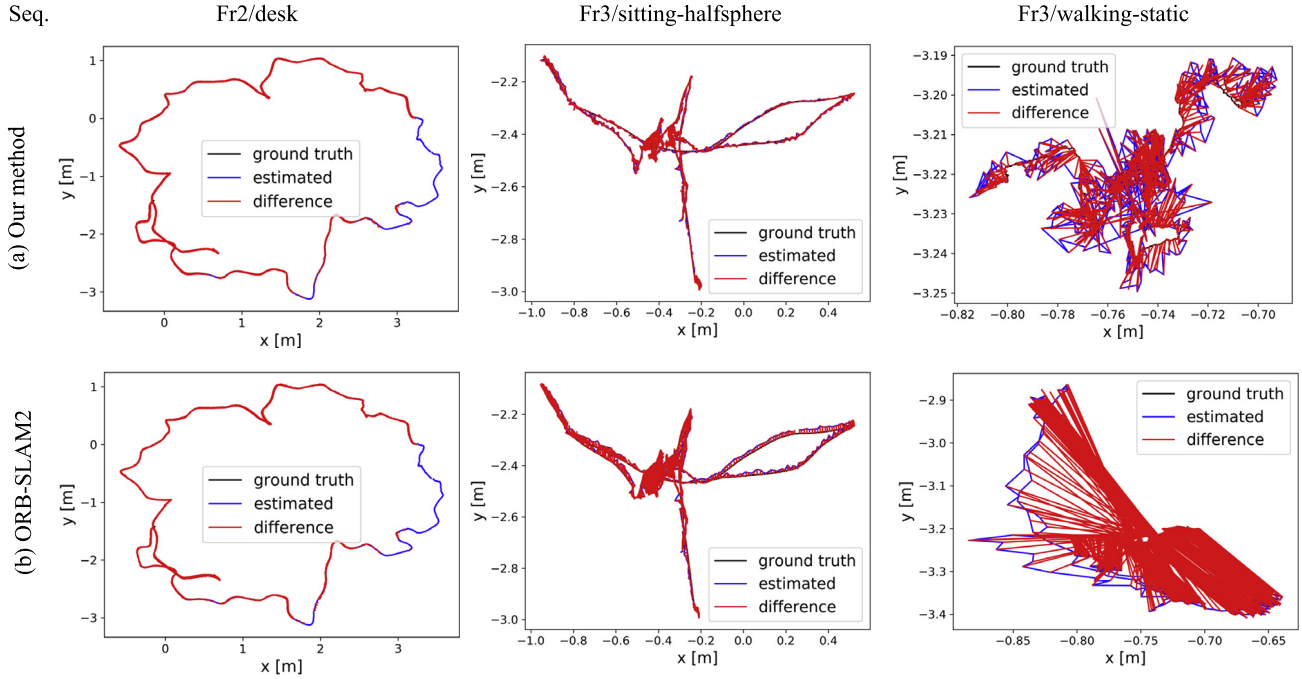


Fig. 10. Kinect trajectories estimated by the proposed VO and ORB-SLAM2. The proposed method maintains more robust localization than ORB-SLAM2. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the global consistency is calculated with ground truth. Meanwhile, the standard deviation is applied for evaluating stabilities of the approaches. The quantitative results of the root mean squared error (RMSE) and the standard deviation are shown in Table 1. More detailed statistics and distributions of ATE obtained by the evaluation tool [28] are shown in Fig. 9. The estimated trajectories are shown in Fig. 10. The first row of Fig. 10 shows the trajectories estimated by our robust approach. Meanwhile, the trajectories estimated by ORB-SLAM2 are shown in the second row. The shorter the red lines in Fig. 10, the better the localization.

The two methods obtain similar localizations in static environments that do not contain any dynamic factors. Compared with

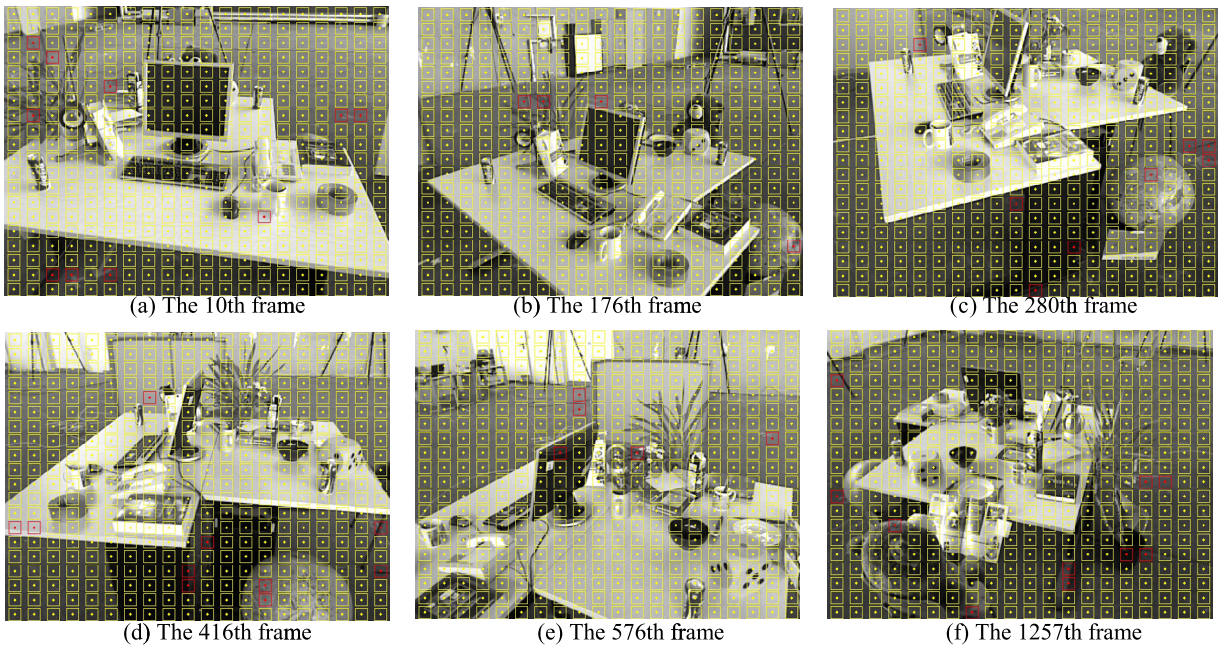


Fig. 11. Area state detection in static environments from "Fr2/desk" sequence. Only a few areas are wrongly regarded as dynamic areas.

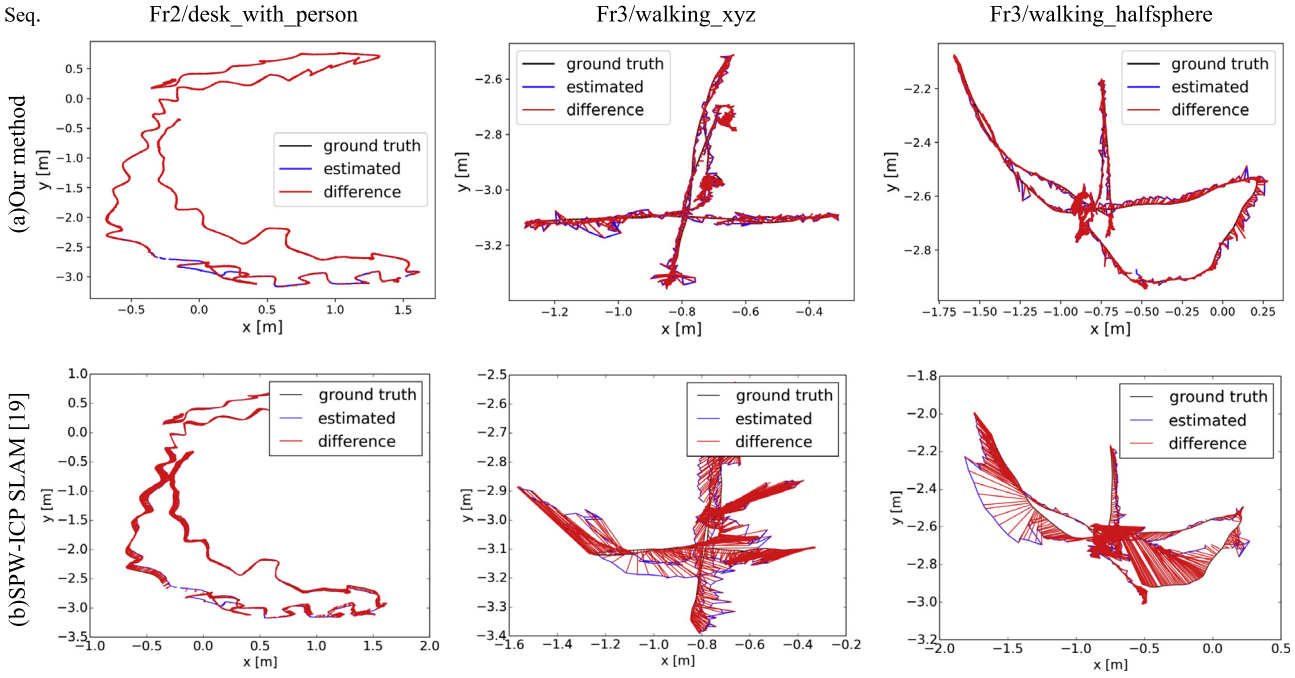


Fig. 12. Kinect trajectories estimated by the proposed VO and SPW-ICP SLAM.

ORB-SLAM2, the increase in the RMSE does not exceed 10% in static environments. The reason for the performance degradation is that the proposed method wrongly eliminates static ORB features according to the area state detection. Fig. 11 shows that only a few areas are wrongly determined as dynamic areas in "Fr2/desk". On average, only 6.04 ORB features are determined as dynamic features for each frame in the sequence. From another perspective, the proposed algorithm is nearly able to position the camera accurately as ORB-SLAM2 in static environments. However, with the dynamic factors, the performance of ORB-SLAM2 obviously degrades. Especially in the "Fr3/walking-static" sequence, a moving person with a checked shirt provides a large number of dynamic features. Those features confuse the reasons for the movements of all features, which seriously break the assumption of static environments. Compared with ORB-SLAM2, the proposed algorithm obtains a 98.16% improvement in the RMSE for the "Fr3/walking-static" sequence. Moreover, this robust approach is more stable than ORB-SLAM2 in dynamic environments, achieving 46.76% and 97.81% standard deviation improvements for the low-dynamic and highly dynamic environments, respectively.

Table 1
ATE [m] of ORB-SLAM2 and the proposed method in static and dynamic environments.

Seq.		ORB-SLAM2		Proposed method		Improvement	
		RMSE	Standard deviation	RMSE	Standard deviation	RMSE	Standard deviation
Static	Fr1/desk	0.0160	0.0090	**0.01487**	**0.0086**	**7.06**%	**4.44**%
	Fr1/desk2	**0.0237**	**0.0117**	0.02461	0.0127	−3.84%	−8.55%
	Fr1/room	**0.0934**	**0.0196**	0.0989	0.0254	−5.89%	−29.60%
	Fr2/desk	**0.0128**	**0.0060**	0.0135	0.0052	−5.47%	**13.3**%
	Fr2/xyz	**0.0035**	**0.0017**	0.0036	**0.0017**	−2.86%	**0**%
	Fr3/office	**0.0140**	**0.0064**	**0.0140**	0.0070	**0**%	−9.38%
	Fr3/nst	**0.0192**	**0.0076**	0.0208	0.0103	−8.33%	−35.53%
Low-dynamic	Fr3/sitting-halfsphere	0.0264	0.0139	**0.0148**	**0.0074**	**43.94**%	**46.76**%
Highly dynamic	Fr3/walking-static	0.4181	0.1828	**0.0077**	**0.0040**	**98.16**%	**97.81**%

Table 2
RPE of DVO, BaMVO, SPW-IAICP and the proposed method.

Seq.		RMSE of translational drift [m/s]				RMSE of rotational drift [°/s]			
		DVO [17]	BaMVO [26]	SPW-IAICP [19]	Proposed method	DVO [17]	BaMVO [26]	SPW-IAICP [19]	Proposed method
Static	Fr2/desk	0.0296	0.0299	0.0173	**0.0072**	1.3920	1.1167	0.7266	**0.4506**
	Fr3/long-office	0.0231	0.0332	0.0168	**0.0084**	1.5689	2.1583	0.8012	**0.4627**
Low-dynamic	Fr2/desk-person	0.0354	0.0352	0.0173	**0.0068**	1.5368	1.2159	0.8213	**0.4359**
	Fr3/sitting-static	0.0157	0.0248	0.0231	**0.0089**	0.6084	0.6977	0.7228	**0.2809**
	Fr3/sitting-xyz	0.0453	0.0482	0.0219	**0.0109**	1.4980	1.3885	0.8466	**0.4717**
	Fr3/sitting-rpy	0.1735	0.1872	0.0843	**0.0499**	6.0164	5.9834	5.6258	**0.7830**
	Fr3/sitting-halfsphere	0.1005	0.0589	0.0389	**0.0168**	4.6490	2.8804	1.8836	**0.5690**
Highly dynamic	Fr3/walking-static	0.3818	0.1339	0.0327	**0.0101**	6.3502	2.0833	0.8085	**0.2571**
	Fr3/walking-xyz	0.4360	0.2326	0.0651	**0.0292**	7.6669	4.3911	1.6442	**0.5847**
	Fr3/walking-rpy	0.4038	0.3584	0.2252	**0.0561**	7.0662	6.3398	5.6902	**1.021**
	Fr3/walking-halfsphere	0.2628	0.1738	0.0527	**0.0352**	5.2197	4.2863	2.4048	**0.7618**

Table 3
ATE [m] of SLAM approaches and the proposed VO approaches.

Seq.	Motion removal + DVO SLAM [17]		SPW-IAICP SLAM [19]		Our VO method		Improvement compared with [19]	
	RMSE	Standard deviation	RMSE	Standard deviation	RMSE	Standard deviation	RMSE	Standard deviation
Fr3/sitting-xyz	0.0482	0.0282	0.0397	0.0206	**0.0087**	**0.0041**	**78.1**%	**80.10**%
Fr3/sitting-halfsphere	0.0470	0.0249	0.0432	0.0246	**0.0148**	**0.0074**	**65.7**%	**69.92**%
Fr2/desk-person	0.0596	0.0239	0.0484	0.0237	**0.0060**	**0.0027**	**87.6**%	**88.61**%
Fr3/walking-static	0.0656	0.0536	0.0261	0.0122	**0.0078**	**0.0040**	**70.1**%	**67.21**%
Fr3/walking-xyz	0.0932	0.0534	0.0601	0.0330	**0.0222**	**0.0122**	**63.1**%	**63.03**%
Fr3/walking-rpy	0.1333	0.0839	0.1791	0.1161	**0.0388**	**0.0241**	**78.3**%	**79.24**%
Fr3/walking-halfsphere	0.1252	0.0903	0.0489	0.7266	**0.0328**	**0.0184**	**32.9**%	**97.47**%

Note that the proposed method is based on indirect methods, therefore, the loop closure of ORB-SLAM2 can be used. This proposed method with the loop closure is tested on the static sequence "Fr2/desk" which contains loops, and shows that the RMSE of ATE decreases from 0.0135 m to 0.0088 m. However, loops have not been detected in dynamic sequences.

4.2. Comparisons with robust VO

Furthermore, to compare the performances of existing VO methods in dynamic environments, more sequences are tested, consisting of static sequences, low-dynamic and highly dynamic sequences. DVO (Dense Visual Odometry) is a kind of direct methods based on RGB-D sensors. It handles dynamic environments with a small quantity of moving objects by a robust weighting function. BaMVO [29] is particularly designed to handle dynamic environments with a background model. SPW-IAICP (Intensity-Assisted Iterative Closest Point with Static Point Weighting) is the VO of the work [19] and provides detailed experimental results on the TUM dataset.

Since most sources of robust VO approaches are closed, we adopt results of the evaluation from their papers. To quantify the odometry drift, the Relative Pose Error (RPE) [22] is adopted. The RMSE values of the translational drift and rotational drift are shown in Table 2. It is clear that, SPW-IAICP achieves better results than DVO and BaMVO for most sequences, and their analyses have been discussed in [19]. Significantly, due to the precise egomotion estimation by minimizing reprojection errors of static features, the proposed method outperforms the three methods mentioned above for all of the sequences in Table 2.

4.3. Comparisons with robust SLAM

Finally, the robust VO in this work is compared with complete SLAM systems of [17] and [19] in dynamic environments. In DVO SLAM, a metrical nearest neighbour search is adopted for candidate keyframes. Then the average entropy and an entropy ratio test are utilized for loop detection, while SPW-IAICP SLAM uses a geometric proximity, a common visible part and a forward backward consistency check to detect loops. The two loop closures employ a pose graph for loop optimization. However, the optimization based on the pose graph does not adjust the environment map. The RMSE and standard deviation of ATE are shown in Table 3. It is worth noting that the proposed VO method still outperforms the two SLAM systems which are designed for dynamic environments. Compared with SPW-IAICP SLAM, the proposed VO method obtains

a 68% improvement in RMSE and a 78% improvement in standard deviation on average. Some of the trajectories are shown in Fig. 12, and the positioning results can be found on GitHub [30].

The proposed VO is implemented on a laptop with Ubuntu 14.04, equipped with an Intel i7-4720HQ CPU (2.60 GHz) and 16 GB RAM. ORB features and edges are synchronously extracted in different child threads. Compared with the tracking thread of ORB-SLAM2, the increased computation costs of the proposed method lie primarily in the edge alignment, which require 6 ms per frame on average. The tracking thread can run at 20 Hz or greater.

5. Conclusion

To overcome the localization accuracy degradation in dynamic environments, a novel VO method based on edges and points is proposed in this work. Dynamic image areas are detected by DT errors from the edge alignment with a reference frame. To handle motion blur in images, an improved strategy of reference frame selection is presented. Static weights from DT errors are added to egomotion estimation based on reprojection errors. Compared with other motion removal approaches, the proposed method focuses on sparse edge alignment without dense segmentation of dynamic areas, which results in real-time operation. Experiments on the TUM dataset show that the method in this work outperforms state-of-the-art approaches. Especially for the highly dynamic sequence "Fr3/walking-rpy", the proposed VO method reduces the RMSE of ATE from 0.1791 m [19] to 0.0388 m, which improves the performance by 78.3%. Compared with traditional VO, the proposed method also achieves comparable performance in static environments. Future work will focus on loop closures in dynamic environments to enhance robustness for a complete SLAM system. And the depth uncertainty of edges introduced by Kinect-like sensors will be taken into consideration for the further improvement.

References

[1] C. Cadena, L. Carlone, H. Carrillo, et al., Simultaneous localization and mapping: Present, future, and the robust-perception age, IEEE Trans. Robot. 32 (6) (2016) 1309–1332.

[2] D. Scaramuzza, F. Fraundorfer, Visual odometry [tutorial] part I: The first 30 years and fundamentals, IEEE Robot. Autom. Mag. 18 (4) (2011) 80–92.

[3] G. Klein, D. Murray, Parallel tracking and mapping for small AR workspaces, in: Proc. IEEE Int. Symp. Mixed Augmented Reality, 2007, pp. 225–234.

[4] R. Mur-Artal, J.D. Tardos, Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras, IEEE Trans. Robot. 33 (5) (2017) 1255–1262.

[5] C. Kerl, J. Sturm, D. Cremers, Robust odometry estimation for RGB-D cameras, in: Proc. IEEE Int. Conf. Robot. Autom, 2013, pp. 3748–3754.

[6] J. Engel, J. Stueckler, D. Cremers, Large-scale direct SLAM with stereo cameras, in: IEEE/RSJ Int. Conf. Intell. Robots Syst, 2015, pp. 1935–1942.

[7] J. Engel, V. Koltun, D. Cremers, Direct sparse odometry, arXiv:1607.02565 [cs.CV].

[8] F. Endres, J. Hess, J. Sturm, et al., 3-D mapping with an RGB-D camera, IEEE Trans. Robot. 30 (1) (2014) 177–187.

[9] R.A. Newcombe, et al., Kinectfusion: Real-time dense surface mapping and tracking, in: Proc. IEEE Int. Symp. Mixed Augmented Reality, 2011, pp. 127–136.

[10] I. Cvišić, I. Petrović, Stereo odometry based on careful feature selection and tracking, in: Proc. European Conf. Mobile Robots (ECMR), 2015, pp. 1–6.

[11] D. Nister, O. Naroditsky, J. Bergen, Visual odometry for ground vehicle applications, J. Field Robot. 23 (1) (2006) 3–20.

[12] F. Schenk, F. Fraundorfer, Robust edge-based visual odometry using machine-learned edges, in: IEEE/RSJ Int. Conf. Intell. Robots Syst, 2017, pp. 1–8.

[13] M. Kuse, S. Shen, Robust camera motion estimation using direct edge alignment and sub-gradient method, in: Proc. IEEE Int. Conf. Robot. Autom, 2016, pp. 573–579.

[14] Y. Ling, M. Kuse, S. Shen, Edge alignment-based visual–inertial fusion for tracking of aggressive motions, Auton. Robots (2017) 1–16.

[15] P.F. Felzenszwalb, D.P. Huttenlocher, Distance transforms of sampled functions, Theory Comput. 8 (1) (2012) 415–428.

[16] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395.

[17] D.-H. Kim, S.-B. Han, J.-H. Kim, Visual odometry algorithm using an RGB-D sensor and imu in a highly dynamic environment, Robot Intell. Technol. Appl. 3 (2015) 11–26.

[18] W. Tan, H. Liu, Z. Dong, et al., Robust monocular SLAM in dynamic environments, in: Proc. IEEE Int. Symp. Mixed Augmented Reality, 2013, pp. 209-218.

[19] S. Li, D. Lee, RGB-D SLAM in dynamic environments using static point weighting, IEEE Robot. Autom. Lett. 2 (4) (2017) 2263–2270.

[20] R.K. Namdev, A. Kundu, K.M. Krishna, et al., Motion segmentation of multiple objects from a freely moving monocular camera, in: Proc. IEEE Int. Conf. Robot. Autom, 2012, pp. 4092–4099.

[21] Y. Sun, M. Liu, M.Q.-H. Meng, Improving RGB-D SLAM in dynamic environments: A motion removal approach, Robot. Auton. Syst. 89 (2017) 110–122.

[22] J. Sturm, N. Engelhard, F. Endres, W. Burgard, D. Cremers, A benchmark for the evaluation of RGB-D SLAM systems, in: IEEE/RSJ Int. Conf. Intell. Robots Syst, 2012, pp. 573–580.

[23] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, Orb: An efficient alternative to sift or surf, in: 2011 International Conference on Computer Vision, IEEE, 2011, pp. 2564–2571.

[24] P. Dollár, C.L. Zitnick, Fast edge detection using structured forests, IEEE Trans. Pattern Anal. Mach. Intell. 37 (8) (2015) 1558–1570.

[25] S. Xie, Z. Tu, Holistically-nested edge detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1395-1403.

[26] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 6 (1986) 679–698.

[27] ORB-SLAM2 website. [Online]. Available: https://github.com/raulmur/ORB_SLAM2.

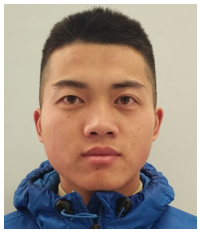
[28] EVO website. [Online] Available: https://github.com/konanrobot/evo.

[29] D.-H. Kim, J.-H. Kim, Effective background model-based RGB-D dense visual odometry in a dynamic environment, IEEE Trans. Robot. 32 (6) (2016) 1565–1573.

[30] Robust VO website. [Online] Available: https://github.com/familyyao/Results_robust_VO.

Erliang Yao received his B.S. degree in 2008 and his M.Sc. degree in 2012 from High-Tech Institute of Xi'an, Shaanxi, China. Now He is a Ph.D. candidate of High-Tech Institute of Xi'an. His latest research interests include robotics vision, path planning and applications to mobile robotics.

Hexin Zhang received his M.Sc. degree in 1994 from Northwestern Polytechnical University, and received his Ph.D. degree in 2001 from High-Tech Institute of Xi'an, Shaanxi, China. Now he is a doctoral supervisor and a professor of High-Tech Institute of Xi'an. His research interests include nonlinear control, fault tolerant control and computer vision.



Hui Xu received his B.Sc. degree in 2016 from High-Tech Institute of Xi'an, Shaanxi, China. Now he is a postgraduate student in High-Tech Institute of Xi'an. His main research interest is robot vision navigation.

Haitao Song received his B.S. degree in 2005 and his M.Sc. degree in 2008 from High-Tech Institute of Xi'an, Shaanxi, China. He received his Ph.D. degree in 2015 from Tsinghua University, Beijing, China. Now he is a lecturer at High-Tech Institute of Xi'an. His current research interests include system modelling, nonlinear control and intelligent decision.

Guoliang Zhang received his Ph.D. degree in 2003 from High-Tech Institute of Xi'an, Shaanxi, China. He was a doctoral supervisor and a professor of High-Tech Institute of Xi'an in 2009. He has published three books, and about 50 articles. Now he is a professor of Chengdu University of Information Technology, Sichuan, China. His latest research interests include advanced control, environment modelling, multi-sensor fusion and autonomous navigation.