ADVANCED ROBOTICS
2019, VOL. 33, NO. 12, 576–589
https://doi.org/10.1080/01691864.2019.1610060

RSJ Taylor & Francis
Taylor & Francis Group

FULL PAPER

Improving monocular visual SLAM in dynamic environments: an optical-flow-based approach

Jiyu Cheng[a], Yuxiang Sun[b] and Max Q.-H. Meng[a]

[a]Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, N.T. Hong Kong SAR, People's Republic of China;
[b]Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong, People's Republic of China

ABSTRACT

Visual Simultaneous Localization and Mapping (visual SLAM) has attracted more and more researchers in recent decades and many state-of-the-art algorithms have been proposed with rather satisfactory performance in static scenarios. However, in dynamic scenarios, the performance of current visual SLAM algorithms degrades significantly due to the disturbance of the dynamic objects. To address this problem, we propose a novel method which uses optical flow to distinguish and eliminate the dynamic feature points from the extracted ones using RGB images as the only input. The static feature points are fed into the visual SLAM system for the camera pose estimation. We integrate our method with the original ORB-SLAM system and validate the proposed method with the challenging dynamic sequences from the TUM dataset and our recorded office dataset. The whole system can work in real time. Qualitative and quantitative evaluations demonstrate that our method significantly improves the performance of ORB-SLAM in dynamic scenarios.

ARTICLE HISTORY
Received 25 September 2018
Revised 14 February 2019
Accepted 13 April 2019

KEYWORDS
Visual SLAM; optical flow; dynamic environments

1. Introduction

For several decades, visual Simultaneous Localization and Mapping (visual SLAM) has achieved significant development. Many state-of-the-art algorithms [1–4] have been proposed with rather satisfactory performance. Given collected information by sensors from the environments, visual SLAM is capable of localizing the mobile robot and generating a representation of the environment as a 3D point cloud.

With the visual SLAM technology getting mature, the research focus has stepped into the robustness age. It requires that the visual SLAM system can work efficiently in challenging environments. Off-the-shelf visual SLAM systems can perform well under the assumption of registering a static environment. In fact, most of the application scenarios contain dynamic objects such as human beings. Once the environment is changing, the localization and mapping capabilities are easy to obtain erroneous alignments (local minima problem) and tracking loss. We call this issue the dynamic SLAM problem.

One solution to solve the dynamic SLAM problem is by information fusion from different sensors [5]. However, information fusion requires additional sensors, which is not a cost-effective way. A popular solution is to use depth information [6, 7] from an embedded IR camera. Usually a segmentation [8] process is adopted. The disadvantage is that RGB-D cameras might not work in outdoor environments due to a limited depth detection (usually between 5 cm and 6 m). In most cases, a monocular camera is more available and portable. Therefore, in this paper, we focus on the problem about how to deal with dynamic SLAM problem using only a monocular camera.

Low cost, easy calibration and portability make the monocular camera a very popular sensor for visual SLAM [9–15]. Many visual SLAM systems use features such as Scale-invariant Feature Transform (SIFT) [16], Speeded Up Robust Features (SURF) [17] or Oriented FAST and Rotated BRIEF (ORB) [18] as an intermediate representation for the raw sensor measurements. This kind of system is known as the feature-based visual SLAM system. In a typical feature-based visual SLAM system, a feature matcher firstly finds feature correspondences between two consecutive frames. Then, the initial camera poses are generated by transformation estimation algorithms, such as Perspective-n-Point (PnP) [19] or motion model like in [20] or registration strategies [21]. After achieving the initial camera poses, a graph which connects camera poses and landmarks is set up and graph solvers like g2o [22] are used to optimize

CONTACT Max Q.-H. Meng ✉ max.meng@cuhk.edu.hk

© 2019 Informa UK Limited, trading as Taylor & Francis Group and The Robotics Society of Japan

the camera poses. The transformation matrix between two frames represents the camera motion. However, in dynamic scenarios, part of the feature points can be extracted from the dynamic objects, object motion will bring noise into the correspondence. As a result, the transformation matrix can represent the camera motion erroneously.

To address this problem, we propose to distinguish and eliminate dynamic feature points from input frames using optical flow in a feature-based monocular SLAM system. The inputs of our system are RGB images obtained by a monocular camera, which is available for almost all robotic platforms. A novel method to distinguish dynamic points using optical flow is introduced, and an efficient dynamic points elimination strategy is applied. The correspondences between two consecutive frames are reliably selected to estimate the camera motion. We have validated our algorithm in both, public TUM dataset and our recorded office dataset. Experiments results demonstrate the improved performance of our method. Our system can work in real time, which is a very important criterion in different tasks in robotics, such as autonomous navigation [23], localization and exploration [24]. This paper is an extension of [25].

The novelties of our work are summarized as follows:

(1) We have proposed a novel method to distinguish dynamic points using optical flow in real time.
(2) We have integrated the proposed method into a feature-based monocular SLAM system. The performance of our method has obtained an outstanding improvement in dynamic scenarios w.r.t ORB-SLAM.
(3) We have recorded a dynamic office dataset and we have tested our proposed method on both, TUM dataset and our office dataset.

This paper is structured as follows. Section 2 describes various existing works in visual SLAM. Section 3 introduces and formulates the problem of performing visual SLAM in dynamic environments. Section 4 provides the details of our proposed method. Experimental results and discussions are shown in Section 5. Finally, we provide a conclusion and the future work.

2. Related work

In order to provide a solution to the visual SLAM problem in dynamic environments, various strategies have been cited. These strategies can be categorized into three main categories.

The first category is based on information fusion [26–28]. Information fusion is a reasonable way to solve the proposed problem, while in some cases additional sensors are not available and processing fusing data may be computational time-consuming. With multi-modal sensor information (e.g. color cameras, depth sensors, LiDAR, IMU), the main advantages of each sensor can be used for estimating more robust and accurate poses. Bloesch et al.[26] combine information between visual and inertial sensors to enable robust performance in dynamic scenarios. The filtering framework uses direct intensity errors as visual measurements within the extended Kalman filter update. The inertial measurements are used to propagate the state of the filter, and the visual information is employed for the filter update. The system exhibits accurate tracking performance and high robustness in roughly and highly unstructured environments. Usenko *et al.* [27] use an Inertial Measurement Unit (IMU) as an additional sensor. They use an energy function to combine photometric and inertial information. By minimizing the energy function, camera pose, velocity and IMU bias are simultaneously estimated. Kim *et al.* [28] use an RGB-D sensor and an IMU to accurately estimate camera trajectory in dynamic environments. They firstly generate 3-D feature points based on SURF descriptor and they use the IMU to compensate rotation of feature points to have the same rigid body transformation matrix between the successive images from the RGB-D sensor. The generated 3-D feature points are divided into dynamic or static feature points using motion vectors and static feature points are used to compute the rigid body transformation matrix.

The second category of approaches is based on the RGB-D camera. In [6, 7, 29–31] depth information is used to detect and eliminate dynamic elements. Kim et al. [29] combine the image information and the 3D position information of the features, where the image information can contribute to precisely detect matched features. Dynamic objects are classified into inliers or outliers depending on dynamically moving features in the image, where outliers are rejected using RANSAC [32]. Wang et al. [30] use the segmentation technique to eliminate dynamic objects. They improved the method in [33] to get a better segmentation result. Consequently, the robustness of their SLAM system has been improved. Kim et al. [31] have proposed a robust background model-based dense-visual-odometry algorithm which can deal with dynamic factors. They firstly warp consecutive depth images to equalize the viewpoints. From the differences between consecutive pairs in the warped depth image, the background image is estimated using a nonparametric model. Sun *et al.* [6] also use segmentation to discard dynamic objects. They firstly compute the transformation matrix between two consecutive frames, then they use ego-motion compensation to get a differencing frame. The pixel value of the coordinates shows whether there is motion or not. A particle filter-based tracking helps to distinguish dynamic elements in each frame. Finally, they use vector quantization-based segmentation to segment dynamic objects out and remaining information will be the input of the system. Li et al. [7] use only edge points to conduct visual odometry based on frame-to-keyframe registration. They give each extracted edge point a static weight which indicates the likelihood of one point being static or dynamic. The static weight is added into an intensity assisted iterative closest point method to perform the registration task. The proposed method can work in real time. Despite the real-time SLAM performance, RGB-D sensors are not suitable for working in outdoor scenarios due to its limited depth detection, which is a crucial drawback of this category of methods.

The third category of approaches are based on monocular camera [34–37]. Bleser *et al.* [34] relies on a CAD model of the object in the scene to initialize the first camera pose and to obtain 3D positions for features on the model. Scene features are tracked from frame to frame and reconstructed in 3D automatically which makes tracking feasible in the environments. Imre et al. [35] use a hybrid static-moving camera setup to conduct pose estimation. Static cameras are used to build a sparse 3D model of the scene. The pose of the moving camera is estimated with respect to the sparse model. Shimamura *et al.* [36] propose to segment outliers of feature points. They detect features on moving objects by building an angle histogram based on outliers, and by using the EM [38] algorithm to estimate parameters for the GMM is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. Tan et al. [37] propose a novel method to represent the dynamic environment. The proposed method can sense the change generated from motion or occlusion. The core of the method is the updating of the keyframes. If there is a big changed part, the keyframe will be replaced to ensure the changed part will not bring into error for the localization.

3. Problem statement

3.1. Notation

We briefly define the notations used throughout our paper.

The RGB image collected at timestamp j is denoted with $I_j : \omega \subset R^2 \mapsto R$, where ω is the image domain. A 3D point $\mathbf{P} = (x, y, z)^T$ maps to the image coordinates $\mathbf{q} = (x, y)^T$ through the camera projection model $\pi : \mathbb{R}^3 \mapsto \mathbb{R}^2$:

$$\mathbf{q} = \pi(\mathbf{T}, \mathbf{P}), \tag{1}$$

where $\mathbf{T} \in SE(3)$ denotes camera pose. The projection π is determined by the intrinsic camera parameters which are known from calibration. u denotes the matched keypoint on the image corresponding to $\mathbf{P}$. The error term for the observation of a map point $\mathbf{P}_i$ in an image j is

$$e_{i,j} = u_{i,j} - \pi(\mathbf{T}_{iw}, \mathbf{P}_i), \tag{2}$$

where w stands for the world reference. We use $\mathbf{P}^*$ and u^* to denote a dynamic 3D point and the corresponding keypoint on the image.

3.2. Problem formulation

Monocular SLAM relies on a map for localization [39]. For instance, in ORB-SLAM [20], the camera pose is obtained by aligning the current frame w.r.t the previous frame or via global relocalization. Bundle Adjustment (BA) [40] is used to optimize the camera pose based on a feature map.

The problem to be solved in BA can be formulated as follows:

$$\begin{aligned} &\arg\min \sum_{i=1}^{n}\sum_{j=1}^{m} w_{i,j}(u_{i,j} - \mathbf{q}_{i,j})^2 \\ &= \underset{\mathbf{T}_{iw},\,\mathbf{P}_i}{\arg\min} \sum_{i=1}^{n}\sum_{j=1}^{m} w_{i,j}(u_{i,j} - \pi(\mathbf{T}_{iw}, \mathbf{P}_i))^2, \end{aligned} \tag{3}$$

where we assume that there are n 3D points in a local map that can be observed in m views. If a point i can be projected on image i, then w_{ij} is 1 otherwise w_{ij} is set to 0. Iteratively minimizing the cost function the optimal camera pose $\mathbf{T}_{iw}$ can be computed. Figure 1(a) shows Bundle Adjustment in static environments.

However, same keypoints may have a different location in dynamic environments. As shown in Figure 1(b), $\mathbf{P}_2$ is a dynamic 3D point, and moves from $\mathbf{P}_2$ to $\mathbf{P}_2^*$. As a result, the corresponding keypoint on I_j moves from u_2' to $u_2'^*$. The error term for the observation of a map point $\mathbf{P}_i$ in an image j is

$$e_{i,j} = u_{i,j}^* - \pi(\mathbf{T}_{iw}, \mathbf{P}_i). \tag{4}$$

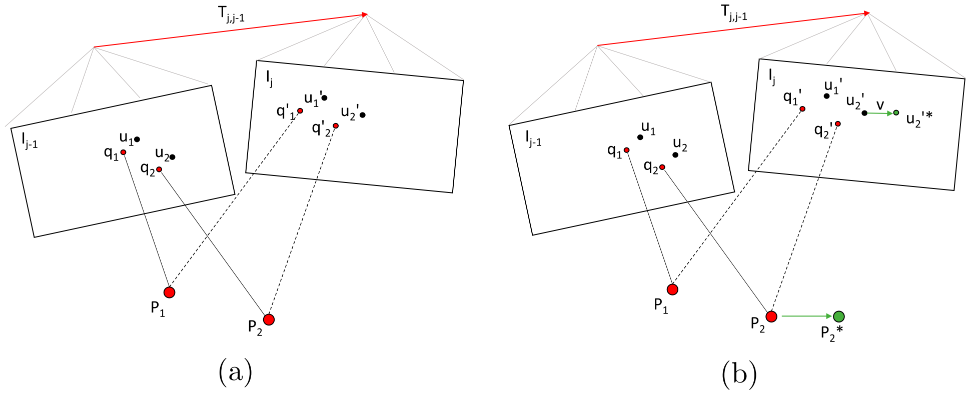


Figure 1. (a) Bundle Adjustment in static environments. (b) Bundle Adjustment in dynamic environments. The red point **P** and green point **P*** represent the static and dynamic 3D point in the space. *u* and *u** are the corresponding point in the image with **P** and **P***. *q* represents the estimated point in the image corresponding with **P**. The figures are best viewed in colour.

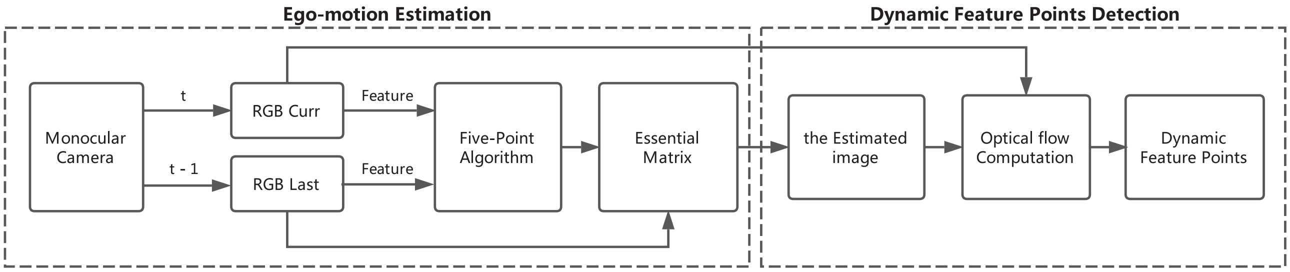


Figure 2. The overview of the proposed method. It consists two modules: *Ego-motion Estimation* and *Dynamic Feature Points Detection*. The *Estimation* module is to estimate the camera motion between two consecutive frames. The *Detection* module is to distinguish dynamic feature points based on optical flow value for each feature point.

Let the vector $v_{i,j}$ denote changes of the location between $u_{i,j}$ and $u^*_{i,j}$, and have modified (3) as follows:

$$\begin{aligned}\arg\min &\sum_{j=1}^{m} w_{i,j}(u^*_{i,j} - \mathbf{q}_{i,j})^2 \\ = \arg\min &\sum_{i=1}^{n}\sum_{j=1}^{m} w_{i,j}(u_{i,j} + \beta_{i,j}v_{i,j} - \mathbf{q}_{i,j})^2 \\ = \arg\min_{\mathbf{T}_{iw},\mathbf{P}_i} &\sum_{i=1}^{n}\sum_{j=1}^{m} w_{i,j}(u_{i,j} + \beta_{i,j}v_{i,j} - \pi(\mathbf{T}_{iw},\mathbf{P}_i))^2,\end{aligned} \tag{5}$$

where

$$\begin{aligned}\beta_{i,j} &= 0, \quad \text{if } u_{i,j} \text{ is static}, \\ \beta_{i,j} &= 1, \quad \text{if } u_{i,j} \text{ is dynamic},\end{aligned} \tag{6}$$

and the other variables have the same definitions with those in (3). If the motion model of the moving objects is given [41], the value of $v_{i,j}$ can be computed and the accuracy of the localization can be ensured by (5). However, if the motion model is unknown, Equation (5) may not lead to a comparable optimization result due to the uncertainty of $v_{i,j}$. Consequently, BA cannot optimize camera poses efficiently in dynamic environments.

4. Methodology

4.1. Method overview

For a keypoint $u_{i,j-1}$ in image $j-1$, the estimated location of its corresponding keypoint $\mathbf{q}_{i,j}$ in frame j can be calculated by:

$$\mathbf{q}_{i,j} = \overline{u_{i,j-1}}\mathbf{T}_{j,j-1}, \tag{7}$$

where $\overline{u_{i,j-1}}$ is the homogeneous representation and $\mathbf{T}_{j,j-1}$ denotes the camera motion between frame j and $j-1$. If $\mathbf{P}_i$ is dynamic, a difference between $\mathbf{q}_{i,j}$ and $u_{i,j}$ will be obtained. We use this difference to determine whether a point is dynamic or not. The core of our proposed method is to eliminate the dynamic element $\beta_{i,j}v_{i,j}$ proposed in (5).

An overview of our proposed method is shown in Figure 2. There are two modules in the proposed method. The first module called *Ego-motion Estimation* is to estimate the camera ego-motion $\mathbf{T}_{j,j-1}$ between two consecutive frames j and $j-1$. Our proposed method works as follows: Firstly, two consecutive images are captured and denoted as RGB Curr and RGB Last, as shown in Figure 2.

We employ the Five-Point Algorithm [42] to estimate the motion of the camera from the last image to the current image by computing the essential matrix. Finally we multiply the last image with the estimated transformation matrix to get a new image which we call the estimated image. In this case, points in the estimated image are converted to the current image. The second module, *Dynamic Feature Points Detection*, calculates optical flow value for each feature point extracted from the current image between the current image and the estimated image and detects dynamic feature points for the current image based on optical flow values. Static points are used for further camera pose estimations.

4.2. Ego-motion estimation

In the *Ego-motion Estimation* module, the inputs are two consecutive frames: *RGB Curr* and *RGB Last* which represent the current frame and the last frame, respectively. We extract the feature points from two input frames, some of which are generated on the moving objects. We denote two feature points sets as $S_1 = (p^1_{d1}, p^1_{d2}, \ldots, p^1_{s1}, p^1_{s2} \cdots\cdots)$ and $S_2 = (p^2_{d1}, p^2_{d2}, \ldots, p^2_{s1}, p^2_{s2} \cdots\cdots)$. As well as in ORB-SLAM [20] ORB features are used as the feature type for pose estimation, and an Octree is used to ensure that feature points can be extracted uniformly from the image.

With the feature points from two frames, we find the correspondence feature pairs C between them. We adopt RANSAC to choose five pairs from all the feature pairs. Note that, some of the pairs are dynamic feature correspondences. For the purposes of this paper, we have an assumption that for all the feature points, we have more static ones than dynamic ones. In this case, the dynamic pairs are regarded as outliers and rejected. By using the RANSAC algorithm, we can ensure that we get all the five correspondences from static feature points. The five-point algorithm uses the five feature pairs to compute the Essential matrix E. The rotation R and translation t can be recovered from E through singular value decomposition (SVD) and disambiguation process [43].

4.3. Dynamic feature points detection

Optical flow is an algorithm to detect object motion which has been extensively studied over the past decades. Given that the coordinate of point A in frame t is (x_1, y_1), if we can find its new place (x_2, y_2) in the following frame t+1 then the motion of point A can be represented as:

$$(\mathbf{u_x}, \mathbf{v_y}) = (x_2, y_2) - (x_1, y_1), \tag{8}$$

where $(\mathbf{u_x}, \mathbf{v_y})$ is a vector that contains the motion direction and distance information.

Optical flow needs to solve a problem to find the coordinate (x_2, y_2), which is formulated in:

$$F(d_x, d_y) = \sum_{(x,y)\in N} [I(x, y, t-1) - I(x + d_x, y + d_y, t)]^2, \tag{9}$$

where $F(d_x, d_y)$ is a cost function. N is the number of neighbor pixels of the center pixel (x_0, y_0) of the patch of pixels, $I(x, y, t-1)$ is the pixel intensity value of point (x, y) in frame $t{-}1$, and $I(x + d_x, y + d_y, t)$ is the intensity value of point$(x + d_x, y + d_y)$ in frame t. Note that here we convert the RGB image to grayscale for further processing.

To optimize the cost function, we can compute the flow vector (d_x, d_y). In *Detection* module, we use Lucas–Kanade [44] to compute optical flow values of feature points extracted from the current image. A predefined tolerance τ is used to determine which point is dynamic by the following inequalities:

$$\begin{aligned} d &> \tau, \quad \text{if } f_i \in F_{\text{dynamic}}, \\ d &< \tau, \quad \text{if } f_i \in F_{\text{static}}, \end{aligned} \tag{10}$$

where $d = \sqrt{d_x^2 + d_y^2}$ is the L_2 norm of the flow vector for feature point f_i, and F_{dynamic} and F_{static} are dynamic and static feature points sets, respectively. The tolerance parameter τ can be updated using the following formulation:

$$f(t, R) = r \cdot \exp(t^2), \tag{11}$$

where $f(t, R)$ represents tolerance τ. R is the rotation of the camera and t is the translation of the camera. As we can see, if R or t gets larger, the value of τ will get larger.

Through *Ego-motion Estimation*, we get the transformation matrix T between the last image and the current image. Then we multiply the last image with the estimated transformation matrix to get the warped image. Points in the estimated image are converted to the same coordinate system as those in the current image. We compute the optical flow values of feature points extracted from the current frame. Then we distinguish dynamic points based on the optical flow values. Figure 3 shows the dynamic feature points result of our method.

4.4. Integration with ORB-SLAM

ORB-SLAM [20] adopts a local feature map to optimize the camera poses. The feature points in the map are generated by triangulating ORB features from connected keyframes in the covisibility. Given each unmatched ORB in keyframe K_i, the system will search for a match with the other point in other keyframes. When in dynamic

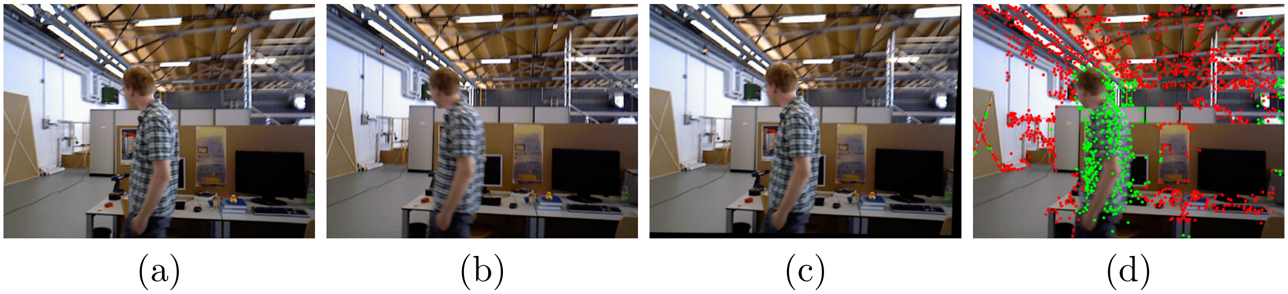


Figure 3. An experimental result of our method. (a)–(c) represent the previous image, the current image and the estimated image. (d) is the result of our method. Red points are static and green points are dynamic. The figures are best viewed in colour.

environments, dynamic feature points will be generated from the moving points and added to the feature map. In this case, the map is not reliable enough to ensure the accuracy of pose estimation.

The integration process can be shown in Algorithm 1. For each input RGB image, we check if it should be inserted as a keyframe. We conduct ego-motion estimation and dynamic feature points detection. For feature-based visual SLAM, one big challenge is that when the system cannot get enough feature points, the tracking may get lost. This is because the feature points in the current frame cannot find a correspondence in the local feature map. In this case, Bundle Adjustment cannot work. To avoid this phenomenon, we first sort the optical flow values. Then we keep the N feature points p_i corresponding with the top N smallest optical flow values. For the rest of the feature points, we use the method in *Dynamic Feature Points Detection* to judge if p_i is a dynamic feature point. If so, we will delete it. And the local feature map will be updated using the process keyframes. In this way, we can ensure that: (1) the tracking of camera poses will not get lost since we have enough feature landmarks; (2) the tracking accuracy is high since for each update the local feature map contains few dynamic feature points.

5. Experiment

In this section, we demonstrate the feasibility and effectiveness of our method by using the public TUM RGB-D dataset and our recorded dynamic office dataset. In the experiments, we integrate our method into the visual SLAM system. The proposed method is to eliminate dynamic feature points which may degrade the performance of visual SLAM. In this study, we adopt the ORB-SLAM [20] as the visual SLAM scheme, which is a state-of-the-art feature-based monocular SLAM system. For ORB-SLAM, we have changed the feature extraction part and our algorithm preprocesses the input data of the feature matching module.

```
Input: the previous keyframe $K_i$, image sequence $H$
Output: Local feature map $M$
Input an RGB image $I$;
for image number in range of length (H) do
    Check if $I$ can be inserted as a keyframe based
    on the criteria proposed in ? ;
    if $I$ is a keyframe then
        Insert $I$ as a new keyframe $K_{i+1}$ ;
        Extract feature points from $K_i$ and $K_{i+1}$ to
        $S_1, S_2$ ;
        Select five correspondances $C$ between $S_1$ &
        $S_2$ ;
        Compute Essential matrix $E$ ;
        Recover $R$ and $t$;
        $K_{warped} = K_i * T$;
        Compute optical flow values for feature
        points $p_i$ between $K_{warped}$ and $K_{i+1}$;
        Sort the values in a set $V$;
        for point $p_i$ in $V$ do
            if optical flow value $V_i$ not in top $N$
            smallest values then
                if optical flow value $V_i > \tau$ then
                    delete the feature point $p_i$;
                end
            end
        end
        Update the local feature map $M$;
    end
end
final ;
return $M$;
```

Algorithm 1: The integration of proposed method with original ORB-SLAM system

5.1. Experimental setup

We have tested our method on both the base of the public TUM RGB-D dataset [45] and our dynamic office dataset. For each video sequence in the two datasets, we carry out two kinds of experiments. One is the evaluation of the performance of the proposed method. The other is the comparison of the performance of ORB-SLAM without and with integrating our method.

For TUM Dynamic RGB-D dataset, We adopted three types of scenarios for our experiments: *desk*, *sitting* and *walking* scenarios. There are four types of movement: *halfsphere*, *rpy*, *static* and *xyz* of which *halfsphere* means the camera moves on a small half sphere and *rpy*, *static*, *xyz* for rotating along the principal axes, keeping the same orientation, moving along three directions, respectively. In this paper, we will use the words *fr*, *half* , *w*, *s*, *d*, *v* to denote *freiburg*, *halfsphere*, *walking*, *sitting*, *desk*, *validation* in the names of sequences.

For our dynamic office dataset, we conduct the experiments in real time with a handheld camera. In each sequence, there are several camera ego-motions which are compatible with the real world. The sequences are streamed using the ROS openni driver with an Asus Xtion Pro Live. The RGB and depth images are synchronized and associated using the TUM benchmark tool. Detailed characteristics are listed in Table 1.

We use the OptiTrack motion capture system to record the ground truth of camera poses. The motion capture system consists of eight Flex3 cameras. The system is shown in Figure 4. It runs at 100 Hz with a resolution of 640*480. The system provides the sub-millimeter tracking accuracy for the RGB-D camera.

Table 1. Detailed characteristics of the sequences in the recorded dynamic office dataset.

Sequence	Frames	Duration	Frequency	Camera Ego-motion
cu/dy/1	1251	65.3 s	19.2 Hz	xyz + halfsphere
cu/dy/2	487	25.5 s	19.1 Hz	circle
cu/dy/3	342	17.9 s	19.1 Hz	xyz + rpy
cu/dy/4	550	28.3 s	19.4 Hz	random

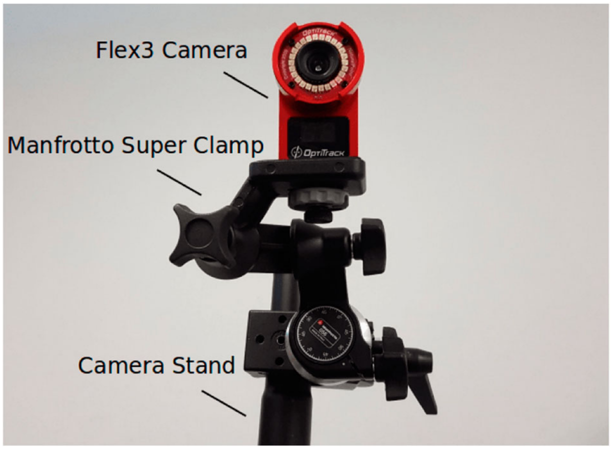


Figure 4. The opti-track motion capture system.

Our experiments are conducted on a PC equipped with an Intel i7 CPU and 16 GB memory. In terms of the ATE plot, for each plot, we implement the experiment for ten times and choose the mean one based on the ATE values.

5.2. Evaluations of proposed method

Figure 5 shows some selected experimental results of the proposed method. For each column, the top one is the original RGB image and the bottom one is the result using the proposed method. Colored points represent feature points extracted from the current image. Red points are static ones and green points are dynamic ones. As we can see, most of the dynamic feature points are on the moving people, and most of the static ones come from the static background. The quantitative results are given in Table 2. We first count the following numbers:

- TP (True Positive): The number of red feature points that belongs to static ones.
- FP (False Positive): The number of red feature points that belongs to dynamic ones.
- TN (True Negative): The number of green feature points that belongs to dynamic ones.
- FN (False Negative): The number of green feature points that belongs to static ones.

We use the following metrics for the quantitative evaluations: False Positive Rate (FPR), False Negative Rate (FNR), Recall (Re), Precision (Pr) and Percentage of Wrong Classifications (PWC). They are calculated as follows,

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}, \tag{12}$$

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{TP}+\mathrm{FN}}, \tag{13}$$

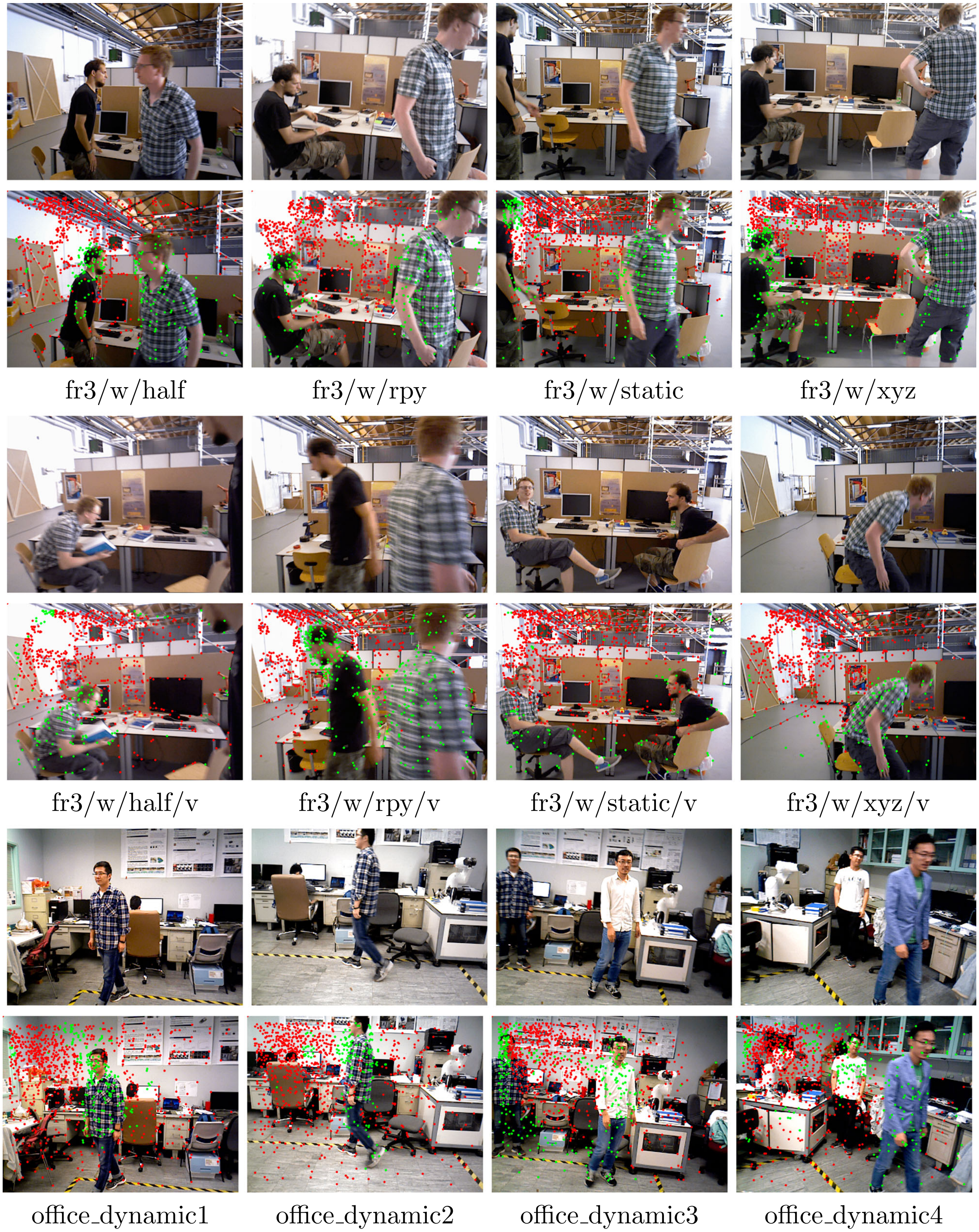
$$\mathrm{Re} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \tag{14}$$

$$\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \tag{15}$$

$$\mathrm{PWC} = \frac{\mathrm{FN}+\mathrm{FP}}{\mathrm{TP}+\mathrm{FN}+\mathrm{FP}+\mathrm{TN}}. \tag{16}$$

We select 12 sequences as our test samples. For each sequence, we randomly select five keyframes and compute the metrics for them. Finally, we take the averages of the metric values for every five keyframes and use them to show the results on each sequence.

From Table 2 we can see that the values of FPR, FNR and PWC are pretty low and the ones of Pr and Re are very


fr3/w/half

fr3/w/rpy

fr3/w/static

fr3/w/xyz

fr3/w/half/v

fr3/w/rpy/v

fr3/w/static/v

fr3/w/xyz/v

office_dynamic1

office_dynamic2

office_dynamic3

office_dynamic4

Figure 5. Selected experimental results of our proposed method. For each column, the top one is the original RGB image and the bottom one is the result using proposed method. Colored points represent feature points extracted from the current image. Red points are static and green points are dynamic. As we can see, our method is able to effectively distinguish dynamic points in dynamic scenarios like these in figure. The figures are best viewed in colour.

Table 2. Results of discrimination for dynamic feature points.

Sequence	FPR (%)	FNR (%)	Re (%)	Pr (%)	PWC (%)
fr3/w/half	2.4	3.0	97.0	99.1	2.7
fr3/w/rpy	3.5	4.6	95.4	98.3	4.3
fr3/w/static	10.3	3.2	96.8	95.8	5.3
fr3/w/xyz	2.6	6.5	93.5	98.8	5.3
fr3/w/half/v	2.4	5.3	94.7	99.5	5.0
fr3/w/rpy/v	0.7	1.7	98.3	99.5	1.3
fr3/w/static/v	9.1	2.2	97.8	98.9	3.0
fr3/w/xyz/v	2.0	3.0	97.0	99.6	2.8
office_dynamic1	9.1	3.4	96.6	98.9	4.0
office_dynamic2	8.3	3.3	96.7	97.4	4.5
office_dynamic3	25.0	6.7	93.3	84.8	14.0
office_dynamic4	11.8	6.0	94.0	97.5	7.0

high. The result shows that our method can efficiently distinguish dynamic feature points from all.

From the result images we can see that, some feature points on dynamic objects are regarded as static ones. This is because when an object is dynamic, not all the parts of it are dynamic. For instance, in fr3/w/static/v sequence, two persons are sitting in a chair, and chatting. In this case, some parts of their body are static, so our algorithm regarded feature points on these parts as static ones. In some images, some static points are taken for a dynamic one, we think there are three reasons. First, it is the ego-motion estimation. Ego-motion is based on the perspective transformation matrix. In highly dynamic scenarios, dynamic feature points usually bring noise into the computation of the transformation matrix which will degrade the performance of our method. The second reason is the noise from the camera motion. Some images in selected sequences are ambiguous due to the camera motion. This ambiguity may impose a negative effect on feature extraction or ego-motion estimation. The third reason is thresholding. We use a threshold to determine whether a point is dynamic, and the threshold is set to 2 in our experiments. However, for different sequences, optimal threshold values may be different. We will try to find a way to update the threshold value based on the sequence conditions.

5.3. Evaluation of visual SLAM

In this part, we evaluate the performance of ORB-SLAM after integrating our proposed method. Both, qualitative and quantitative results are given to demonstrate the feasibility of our method. Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) metrics are adopted to conduct the quantitative evaluation. The metric ATE measures the global consistency and RPE measures the odometry drift.

5.3.1. Qualitative results

Figures 6 and 7 consist of selected ATE plots which show the qualitative results of ORB-SLAM after integrating with our method. The ground truth is represented as the black line, and the estimated trajectory as the blue line, the differences as red lines. In Figures 6 and 7, We use -*with* and -*without* to represent that the experiments are performed with and without our method.

The sequence fr3/w/half, fr3/w/xyz and fr3/w/xyz_v belong to highly dynamic scenarios. The performance of ORB-SLAM is degraded in such scenarios. The estimated camera poses have significant biases with the groundtruth. This is because once the noise is introduced into the system, it will be accumulated. As a result, the camera cannot get an accurate localization. After using our method, the performance improved very significantly. The differences are decreased and the estimated trajectory is aligned with the ground truth much better. The sequence fr3/s/half, fr2/d/person and fr3/s/xyz belong to low-dynamic scenarios. ORB-SLAM performs very well in such a condition. This is because in low-dynamic sequences, dynamic factors occupy respectively smaller region than those of high-dynamic ones. As a result, static feature points will dominate the process of pose estimation. While we can also see the improvement with our method. For instance, at the top of the trajectory in sequence fr3/s/half, the difference between ground truth and the estimated trajectory is reduced after using our method. These experimental results show that our method can deal with the proposed problem in both high-dynamic and low-dynamic scenarios.

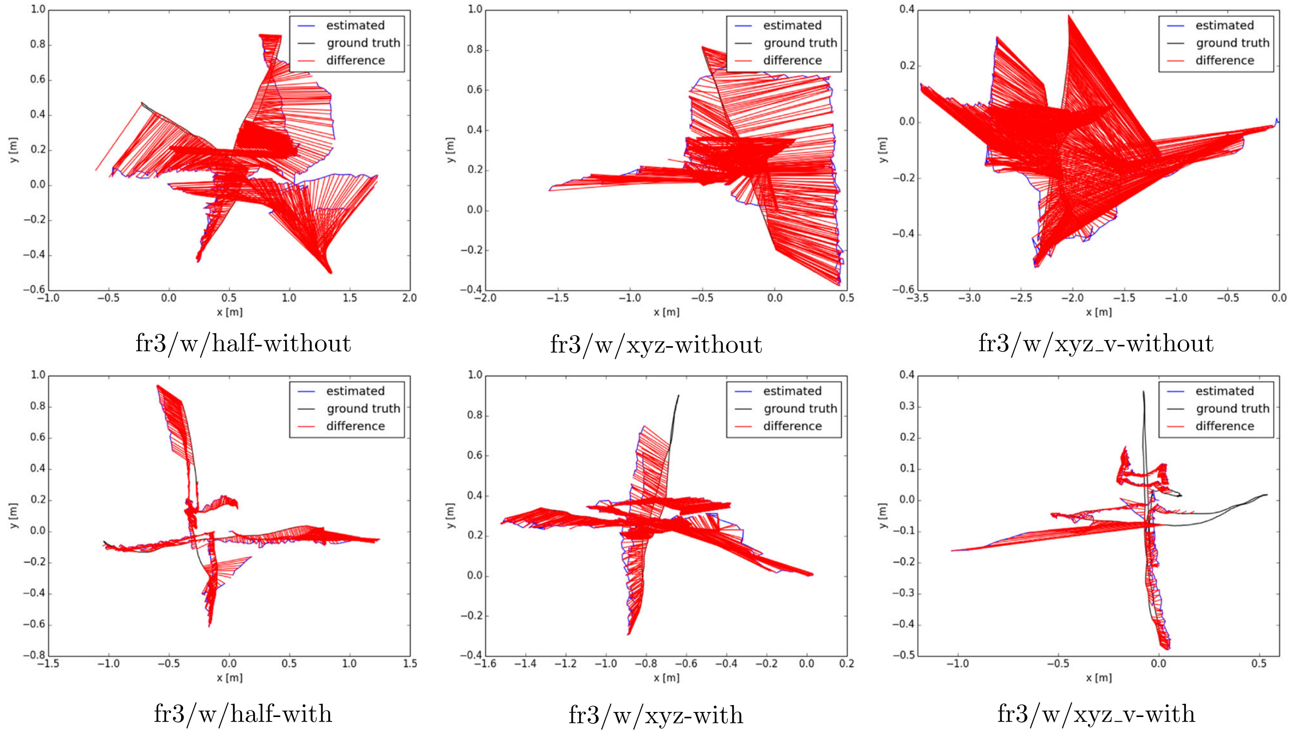
However, in some sequences such as fr3/w/xyz and fr3/w/xyz_v, some camera poses cannot be estimated so that the ground truth cannot be aligned with the estimated trajectory very well. This is because in some cases there are many dynamic feature points. Once we delete them, the system may fail to track the trajectory of the camera due to the default information.

5.3.2. Quantitative results

Tables 3–6 demonstrate the quantitative results of our experiments. *Without Our Approach* means that we use the ORB-SLAM algorithm. *With Our Approach* means that we use the ORB-SLAM algorithm with our method integrated. Root-Mean-Square Error (RMSE), Mean Error, Median Error and the Standard Deviation (S.D.) metrics are used in this paper to help make the analysis.The improvement values [6] in the tables are calculated using

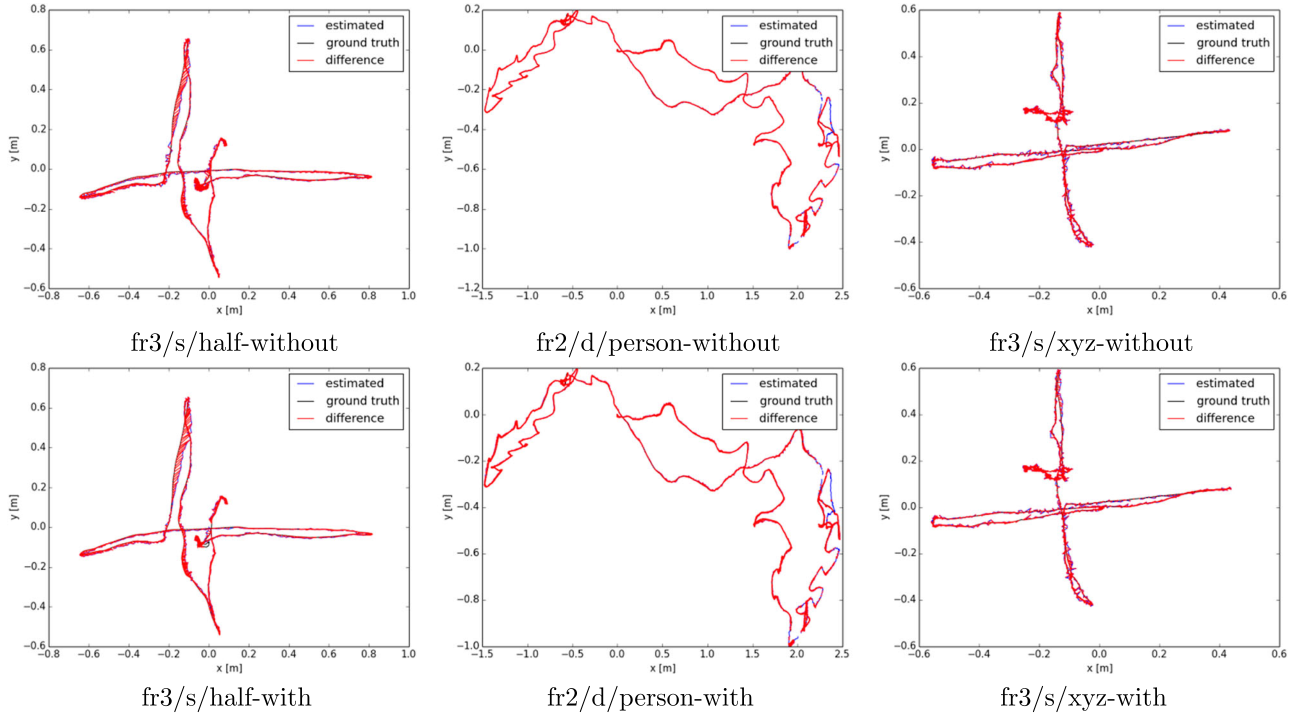
$$\zeta = \left(1 - \frac{\beta}{\alpha}\right) \times 100\%, \qquad (17)$$

where ζ denotes the improvement value; α denotes the value without our method; and β denotes the value with our method. Also, we highlight the RMSE and S.D. values which can reflect the stability of the system.



fr3/w/half-without
fr3/w/xyz-without
fr3/w/xyz_v-without
fr3/w/half-with
fr3/w/xyz-with
fr3/w/xyz_v-with

Figure 6. Plots of ATE for sequences fr3/w/half, fr3/w/xyz/, fr3/s/half.



fr3/s/half-without
fr2/d/person-without
fr3/s/xyz-without
fr3/s/half-with
fr2/d/person-with
fr3/s/xyz-with

Figure 7. Plots of ATE for the sequences fr3/s/half, fr2/d/person, fr3/s/xyz.

Table 3 shows the global consistency performance. As we can see, our method brings significant improvements in all the sequences in terms of RMSE and S.D. For high-dynamic scenarios, the improvements are more obvious and the highest improvement for RMSE is 95.86%. These experimental results demonstrated that our method can deal with the high-dynamic scenarios very effectively. And for the low-dynamic scenarios, our method provides improvements from 5.15% to 25.56%, which is less than that in the high-dynamic scenarios. The reason may be

Table 3. ATE in meters for the experiments without and with our proposed method.

	Without our approach		With our approach		Improvements	
Sequences	RMSE	S.D.	RMSE	S.D.	RMSE (%)	S.D. (%)
fr3/w/half	0.4579	0.2252	**0.1612**	**0.1187**	**64.80**	**52.71**
fr3/w/rpy	0.9046	0.4772	**0.1533**	**0.1119**	**83.05**	**76.55**
fr3/w/xyz	0.4808	0.2011	**0.1899**	**0.1115**	**60.50**	**44.55**
fr3/w/half/v	0.5591	0.3226	**0.0671**	**0.0506**	**88.00**	**84.31**
fr3/w/rpy/v	0.5799	0.4599	**0.0299**	**0.0178**	**95.86**	**96.13**
fr3/w/xyz/v	1.4212	0.6153	**0.1415**	**0.1299**	**90.04**	**78.89**
fr3/s/half*	0.0198	0.0120	**0.0179**	**0.0102**	**9.60**	**15.00**
fr3/s/xyz*	0.0097	**0.0042**	**0.0092**	0.0043	**5.15**	-2.38
fr2/d/person*	0.0090	0.0036	**0.0067**	**0.0029**	**25.56**	**19.44**
office/dynamic1	0.5989	0.2635	**0.0144**	**0.0068**	**97.60**	**97.42**
office/dynamic2	**0.0438**	**0.0192**	0.1479	0.0840	-237.67	-337.5
office/dynamic3	1.1939	0.4960	**0.0515**	**0.0265**	**95.69**	**94.66**
office/dynamic4	0.0881	0.0444	**0.0349**	**0.0163**	**60.39**	**63.29**

Note: The bold values indicate the best performance.

Table 4. ATE in meters for the comparison between DVO SLAM [3] and our method.

	DVO SLAM		Our approach	
Sequences	RMSE	S.D.	RMSE	S.D.
fr3/w/half	0.5287	0.2260	**0.1612**	**0.1187**
fr3/w/rpy	0.7304	0.2837	**0.1533**	**0.1119**
fr3/w/xyz	0.5966	0.2672	**0.1899**	**0.1115**
fr3/w/half/v	0.3735	0.2019	**0.0671**	**0.0506**
fr3/w/rpy/v	0.9115	0.2588	**0.0299**	**0.0178**
fr3/w/xyz/v	0.8778	0.5158	**0.1415**	**0.1299**
fr3/s/half*	0.0616	0.0324	**0.0179**	**0.0102**
fr3/s/xyz*	0.0505	0.0317	**0.0092**	**0.0043**
fr2/d/person*	0.0853	0.0180	**0.0067**	**0.0029**

Note: The bold values indicate the best performance.

Table 5. Translational drift (RPE) in m/s for the experiments without and with our proposed method.

	Without Our Approach		With Our Approach		Improvements	
Sequences	RMSE	S.D.	RMSE	S.D.	RMSE (%)	S.D. (%)
fr3/w/half	0.7304	0.4410	**0.2511**	**0.1881**	**65.62**	**42.65**
fr3/w/rpy	1.3705	0.7737	**0.2397**	**0.1802**	**82.51**	**76.71**
fr3/w/xyz	0.7353	0.4126	**0.2727**	**0.1909**	**62.91**	**53.73**
fr3/w/half/v	0.8360	0.5420	**0.0954**	**0.0759**	**88.59**	**86.00**
fr3/w/rpy/v	0.8169	0.6098	**0.0435**	**0.0247**	**94.67**	**95.95**
fr3/w/xyz/v	2.0314	1.3632	**0.2127**	**0.1984**	**89.53**	**85.45**
fr3/s/half*	0.0292	0.0181	**0.0268**	**0.0153**	**8.22**	**15.47**
fr3/s/xyz*	0.0146	0.0062	**0.0137**	**0.0061**	**6.16**	**1.61**
fr2/d/person*	0.0352	0.0212	**0.0325**	**0.0188**	**7.67**	**11.32**
office/dynamic1	1.8226	1.0027	**0.2829**	**0.1679**	**84.48**	**83.26**
office/dynamic2	0.5475	0.2673	**0.3883**	**0.2150**	**29.08**	**19.57**
office/dynamic3	1.9781	1.0428	**0.5003**	**0.2818**	**74.71**	**72.98**
office/dynamic4	2.6200	1.5543	**1.0389**	**0.7063**	**60.35**	**54.56**

Note: The bold values indicate the best performance.

Table 6. Rotational drift (RPE) in deg/s for the experiments without and with our proposed method.

	Without Our Approach		With Our Approach		Improvements	
Sequences	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
fr3/w/half	17.9510	10.4139	**2.2485**	**1.3470**	**87.47%**	**87.07%**
fr3/w/rpy	24.2553	14.9178	**4.4660**	**3.2084**	**81.59%**	**78.49%**
fr3/w/xyz	12.9045	7.5109	**4.6565**	**3.6878**	**63.92%**	**50.90%**
fr3/w/half/v	18.1410	11.8257	**1.0185**	**0.5497**	**94.39%**	**95.35%**
fr3/w/rpy/v	16.0730	11.8260	**1.1722**	**0.6363**	**92.71%**	**94.62%**
fr3/w/xyz/v	40.7108	27.4890	**2.5161**	**2.2728**	**93.82%**	**91.73%**
fr3/s/half*	**0.8515**	**0.4035**	0.8750	0.4080	-2.76%	-1.12%
fr3/s/xyz*	0.6273	0.3369	**0.5960**	**0.3098**	**4.99%**	**8.04%**
fr2/d/person*	1.3924	0.6925	**1.2722**	**0.6389**	**8.63%**	**7.74%**
office/dynamic1	117.7939	49.3084	**19.4356**	**11.9035**	**83.50%**	**75.86%**
office/dynamic2	**21.0039**	**12.3521**	21.1655	12.3333	-0.77%	-0.15%
office/dynamic3	36.5262	20.5573	**10.2810**	**5.1186**	**71.85%**	**75.10%**
office/dynamic4	105.5477	50.4041	**60.7810**	**38.4076**	**42.41%**	**23.80%**

Notes: The unit for the median values is rad/s. The bold values indicate the best performance.

that in low-dynamic scenarios, the relatively less dynamic feature points can be easily distinguished, so the original ORB-SLAM can perform very well in such a situation. For our office dynamic dataset, we can also see the improvements, while for office/dynamic2 sequence, our system degrades the performance. This is because the camera moves fast and our method cannot deal with the dynamic feature points deletion work very well. Table 4 shows the comparison between DVO SLAM [3] and our proposed system in terms of ATE. DVO SLAM is one of the state-of-the-art direct visual SLAM systems. The better results are highlighted. From the table we can see that for both high-dynamic and low-dynamic scenarios, our method performs better than DVO SLAM.

Tables 5 and 6 demonstrate the tracking performance. As we can see, the results are in line with the above ATE analysis in Table 3. For the high-dynamic sequences, our method significantly improves the performance of ORB-SLAM. However, in low-dynamic sequences, our method brings less improvement even degrades the performance a little bit. We think the reason is that in low-dynamic scenarios, most of the feature points are static and camera poses can be estimated robustly and accurately, which leaves little space for improvements. For our office dynamic dataset, without our approach, the translational and rotational drifts are very high and even higher than those of TUM high dynamic sequences we used. This demonstrates that our office dynamic dataset contains very challenging scenarios. After using our method, the drifts are decreased significantly. However, the drifts are still high for pose estimation which may be left as future work.

6. Conclusion

In this paper, we propose a novel method to distinguish and eliminate dynamic points using a monocular camera. The only input is RGB image and our method can work in real time. The proposed method can be divided into two modules: ego-motion estimation and optical flow-based detection. After integration with our method, the performance of visual SLAM in dynamic scenarios is significantly improved. We conducted experiments on both TUM dataset and our recorded office dataset. Qualitative and quantitative evaluations demonstrated that our method can deal with both high-dynamic and low-dynamic scenarios. However, our method still presents some limitations. For instance, the threshold which is used to distinguish dynamic points is set to a fixed value, which may not be an optimal value for some sequences. Also, when dynamic objects occupy much space of the image, the number of feature correspondences for camera pose estimation will be reduced which may lead to track-lost. In future work, we will use semantic information [46] to enhance the robustness and accuracy. The threshold will be updated online for different motion modes. Also, when there exist many dynamic objects, the system will choose the static area to extract feature points.

Acknowledgements

The authors would like to thank Zhe Min for helping with the experiments and his suggestions for preparing this manuscript.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This project is partially supported by the Hong Kong Research Grants Council General Research Fund [grant number 14205914], Innovation and Technology Commission Innovation and Technology Fund [grant number ITS/236/15], and Shenzhen Science and Technology Innovation Projects JCYJ20170413161616163 awarded to Max Q.-H. Meng.

Notes on contributors

Jiyu Cheng received his B.A. degree in automation from Shandong University, Jinan, China, in 2015. He is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong. His current research interests include autonomous navigation, active localization, and semantic mapping.

Yuxiang Sun received the bachelor's degree from the Hefei University of Technology, Hefei, China, in 2009, the master's degree from the University of Science and Technology of China, Hefei, in 2012, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2017. He is currently a Research Associate with the Robotics Institute, Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong. His current research interests include mobile robots, autonomous vehicles, deep learning, SLAM and navigation, motion detection, and so on.

Dr. Sun is a recipient of the Best Student Paper Finalist Award at the IEEE ROBIO 2015.

Max Q.-H. Meng received his Ph.D. degree in Electrical and Computer Engineering from the University of Victoria, Canada, in 1992. He joined the Chinese University of Hong Kong in 2001 and is currently Professor and Chairman of Department of Electronic Engineering. He was with the Department of Electrical and Computer Engineering at the University of Alberta in Canada, serving as the Director of the Advanced Robotics and Teleoperation Lab and holding the positions of Assistant Professor (1994), Associate Professor (1998), and Professor (2000), respectively. He is affiliated with the State Key Laboratory of Robotics and Systems at Harbin Institute of Technology and the Honorary Dean of the School of Control Science and Engineering at Shandong University, in China. His research interests include robotics, medical robotics and devices, perception, and scenario intelligence. He has published some 600 journal and conference papers and led more than 50 funded research projects to completion as PI. He has served as an editor of several journals and General and Program Chair of many conferences including General Chair of IROS 2005 and General Chair of ICRA 2021 to be held in Xi'an, China. He is an elected member of the Administrative Committee (AdCom) of the IEEE Robotics and Automation Society. He is a recipient of the IEEE Millennium Medal, a Fellow of the Canadian Academy of Engineering, a Fellow of HKIE, and a Fellow of IEEE.

References

[1] Engel J, Schöps T, Cremers D. LSD-SLAM: large-scale direct monocular SLAM. European Conference on Computer Vision; Zurich, Switzerland. Springer; 2014. p. 834–849.

[2] Mur-Artal R, Tardós JD. ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras. IEEE Trans Robot. 2017;33(5):1255–1262.

[3] Kerl C, Sturm J, Cremers D. Dense visual SLAM for RGB-D cameras. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); Tokyo, Japan. IEEE; 2013. p. 2100–2106.

[4] Whelan T, Salas-Moreno RF, Glocker B, et al. Elasticfusion: real-time dense SLAM and light source estimation. Int J Rob Res. 2016;35(14):1697–1716.

[5] Lv W, Kang Y, Qin J. Indoor localization for skid-steering mobile robot by fusing encoder, gyroscope, and magnetometer. IEEE Trans Syst Man Cybern: Syst. 2017;99:1–13.

[6] Sun Y, Liu M, Meng MQ-H. Improving rgb-d SLAM in dynamic environments: a motion removal approach. Rob Auton Syst. 2017;89:110–122.

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

[8] Sun Y, Liu M, Meng MQ-H. Active perception for foreground segmentation: an RGB-D data-based background modeling method. IEEE Trans Autom Sci Eng. 2019.

[9] Newcombe RA, Lovegrove SJ, Davison AJ. DTAM: dense tracking and mapping in real-time. 2011 IEEE International Conference on Computer Vision (ICCV); Barcelona, Spain. IEEE; 2011. p. 2320–2327.

[10] Forster C, Pizzoli M, Scaramuzza D. SVO: fast semi-direct monocular visual odometry. 2014 IEEE International Conference on Robotics and Automation (ICRA); Hong Kong, China. IEEE; 2014. p. 15–22.

[11] Pascoe G, Maddern W, Tanner M, et al. NID-SLAM: Robust monocular SLAM using normalised information distance. Conference on Computer Vision and Pattern Recognition; Honolulu, Hawaii. 2017.

[12] Gálvez-López D, Salas M, Tardós JD, et al. Real-time monocular object SLAM. Rob Auton Syst. 2016;75: 435–449.

[13] Frost D, Prisacariu V, Murray D. Recovering stable scale in monocular SLAM using object-supplemented bundle adjustment. IEEE Trans Robot. 2018;34(3):736–747.

[14] Luo H, Gao Y, Wu Y, et al. Real-time dense monocular SLAM with online adapted depth prediction network. IEEE Trans Multimedia. 2018;21(2):470–483.

[15] Engel J, Koltun V, Cremers D. Direct sparse odometry. IEEE Trans Pattern Anal Mach Intell. 2018;40(3): 611–625.

[16] Lowe DG. Distinctive image features from scale-invariant keypoints. Int J Comput Vis. 2004;60(2):91–110.

[17] Bay H, Tuytelaars T, Van Gool L. Surf: speeded up robust features. European Conference on Computer Vision; Graz, Austria. Springer; 2006. p. 404–417.

[18] Rublee E, Rabaud V, Konolige K, et al. ORB: an efficient alternative to sift or surf. 2011 IEEE International Conference on Computer Vision (ICCV); Barcelona, Spain. IEEE; 2011. p. 2564–2571.

[19] Lepetit V, Moreno-Noguer F, Fua P. EPnP: an accurate O(n) solution to the PnP problem. Int J Comput Vis. 2009;81(2):155–166.

[20] Mur-Artal R, Montiel JMM, Tardos JD. ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans Robot. 2015;31(5):1147–1163.

[21] Min Z, Wang J, Song S, et al. Robust generalized point cloud registration with expectation maximization considering anisotropic positional uncertainties. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); Madrid, Spain. IEEE; 2018. p. 1290–1297.

[22] Kümmerle R, Grisetti G, Strasdat H, et al. g2o: a general framework for graph optimization. 2011 IEEE International Conference on Robotics and Automation (ICRA); Shanghai, China. IEEE; 2011. p. 3607–3613.

[23] Cheng J, Cheng H, Meng MQ-H, et al. Autonomous navigation by mobile robots in human environments: a survey. 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE; 2018. p. 1981–1986.

[24] Wang C, Cheng J, Wang J, et al. Efficient object search with belief road map using mobile robot. IEEE Rob Autom Lett. 2018;3(4):3081–3088.

[25] Cheng J, Sun Y, Chi W, et al. An accurate localization scheme for mobile robots using optical flow in dynamic environments. 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO); Kuala Lumpur, Malaysia. IEEE; 2018. p. 723–728.

[26] Bloesch M, Omari S, Hutter M, et al. Robust visual inertial odometry using a direct EKF-based approach. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); Hamburg, Germany. IEEE; 2015. p. 298–304.

[27] Usenko V, Engel J, Stückler J, et al. Direct visual-inertial odometry with stereo cameras. 2016 IEEE International Conference on Robotics and Automation (ICRA); Stockholm, Sweden. IEEE; 2016. p. 1885–1892.

[28] Kim D-H, Han S-B, Kim J-H. Visual odometry algorithm using an RGB-D sensor and IMU in a highly dynamic environment. Proc. Int. Conf. Robot. Intell. Technol. Appl; Cham. 2015. p. 11–26.

[29] Kim D-H, Kim J-H. Image-based ICP algorithm for visual odometry using a RGB-D sensor in a dynamic environment. Robot Intelligence Technology and Applications 2012; Cham. Springer; 2013. p. 423–430.

[30] Wang Y, Huang S. Towards dense moving object segmentation based robust dense RGB-D SLAM in dynamic scenarios. 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV); Singapore. IEEE; 2014. p. 1841–1846.

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

[32] Fischler MA, Bolles RC. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM. 1981;24(6):381–395.

[33] Ochs P, Malik J, Brox T. Segmentation of moving objects by long term video analysis. IEEE Trans Pattern Anal Mach Intell. 2014;36(6):1187–1200.

[34] Bleser G, Wuest H, Stricker D. Online camera pose estimation in partially known and dynamic scenes. ISMAR 2006. IEEE/ACM International Symposium on Mixed and Augmented Reality, 2006; Santa Barbara, CA, USA. IEEE; 2006. p. 56–65.

[35] Imre HE, Guillemaut J-Y, Hilton ADM. Moving camera registration for multiple camera setups in dynamic scenes. Proceedings of the 21st British Machine Vision Conference; Aberystwyth, Wales, UK. 2010.

[36] Shimamura J, Morimoto M, Koike H. Robust VSLAM for dynamic scenes. MVA. 2011. p. 344–347.

[37] Tan W, Liu H, Dong Z, et al. Robust monocular SLAM in dynamic environments. 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR); Adelaide, SA, Australia. IEEE; 2013. p. 209–218.

[38] Rabiner LR, Juang B-H. An introduction to hidden Markov models. IEEE ASSP Mag. 1986;3(1):4–16.

[39] Mostegel C, Wendel A, Bischof H. Active monocular localization: towards autonomous monocular exploration for multirotor MAVS. 2014 IEEE International Conference on Robotics and Automation (ICRA); Hong Kong, China. IEEE; 2014. p. 3848–3855.

[40] Triggs B, McLauchlan PF, Hartley RI, et al. Bundle adjustment? A modern synthesis. International Workshop on Vision Algorithms; Cham. Springer; 1999. p. 298–372.

[41] Vo M, Narasimhan SG, Sheikh Y. Spatiotemporal bundle adjustment for dynamic 3D reconstruction. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; Las Vegas, Nevada. 2016. p. 1710–1718.

[42] Nistér D. An efficient solution to the five-point relative pose problem. IEEE Trans Pattern Anal Mach Intell. 2004;26(6):756–770.

[43] Wang J, Zha H, Cipolla R. Coarse-to-fine vision-based localization by indexing scale-invariant features. IEEE Trans Syst Man Cybern Part B (Cybern). 2006;36(2): 413–422.

[44] Baker S, Matthews I. Lucas-Kanade 20 years on: a unifying framework. Int J Comput Vis. 2004;56(3):221–255.

[45] Sturm J, Engelhard N, Endres F. A benchmark for the evaluation of RGB-D SLAM systems. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); Algarve, Portugal. IEEE; 2012. p. 573–580.

[46] Cheng J, Sun Y, Meng MQ-HA dense semantic mapping system based on CRF-RNN network. 2017 18th International Conference on Advanced Robotics (ICAR); Hong Kong, China. IEEE; 2017. p. 589–594.