IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 12, 2020, accepted July 28, 2020, date of publication August 3, 2020, date of current version August 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3013806

EF-Razor: An Effective Edge-Feature Processing Method in Visual SLAM

WEN LIU, YAOKAI MO, JICHAO JIAO, AND ZHONGLIANG DENG, (Senior Member, IEEE)
School of Electronics Engineering, Beijing University of Posts and Telecommunications, Beijing 100089, China

Corresponding authors: Wen Liu (liuwen@bupt.edu.cn) and Yaokai Mo (moyaokai@bupt.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61871054.

ABSTRACT Feature-based visual simultaneous localization and mapping (SLAM) is an effective localization approach for robots in unknown environments. Classic handcrafted features perform well in 2D image matching tasks. However, in the tracking task of SLAM, the region at the edge of the object in the image is often unstable because of the lack of spatial information. In this paper, we refer to the features at the edge of the object as edge-features and propose an effective method to process the edge-features in SLAM named Edge-Feature Razor (EF-Razor) for the above problems. EF-Razor first uses the semantics provided by the object detection YOLOv3 to distinguish edge-features. Through additional constraints on edge-features matching in the tracking process, EF-Razor can effectively reduce the impact of unstable features on the SLAM system. Then, EF-Razor adjusts the information matrix to increase the system's trust in the filtered features. This will make the calculation result of the bundle adjustment more stable. In order to evaluate the proposed method, we integrate EF-Razor to ORB-SLAM2 and perform experiments. The comparison results based on public datasets show the proposed method could effectively reduce the absolute trajectory error by 7%.

INDEX TERMS Edge, features, object detection, simultaneous localization and mapping.

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is defined as the question of a moving sensor platform constructing a representation of its environment on the fly while concurrently estimating its ego-motion [1]. It has been a popular research topic in the last two decades in the computer vision and robotics communities. Visual SLAM usually relied on camera, which is cheap and provides rich information about the environment that allows for robust and accurate place recognition [2]. The rapid development of Visual SLAM in recent years has made it widely used in self-driving cars, unmanned aerial, Augmented Reality (AR) and Virtual Reality (VR) [3]–[5].

According to the information used by the front-end, Visual SLAM can be divided into direct method and indirect method (feature-based method). In the direct method, the front-end of SLAM uses a mass of pixels to solve the motion change of the vision sensor by minimizing the intensity/brightness errors between the projected pixels and the landmarks. However, the direct methods rely on the assumption of photometric invariance. In addition, they are limited by the non-convexity of the intensity of the pixel of the image. Although the direct methods usually achieve a higher accuracy, a higher computational complexity of them restricts themselves to be widely used in practical applications.

In contrast, feature-based methods extract keypoints in the image and calculate the one-to-one matches between the 3D keypoints (landmarks) and the image keypoints [6]. In feature-based method, the one-to-one matches and calculation of the reprojection residuals are two key steps that affect the final positioning result. The matches are mostly resolved based on feature matching methods. Therefore, they have strict requirement for the correct matching ability of the features. On the other hand, the camera motion can be optimized by minimizing the pixel distances between the projected 3D keypoints and the detected keypoints. In the optimization objective, we call the pixel distance the reprojection residuals. And, the optimization is well known as bundle adjustment (BA).

The associate editor coordinating the review of this manuscript and approving it for publication was Abdel-Hamid Soliman.

A classic representative of the early work of the feature point method is PTAM [4]. PTAM had two parallelized threads computing motion estimation and mapping, which allows robust state estimation in real-time for the first time. ORB-SLAM [2] is an open-source SLAM algorithm developed by MurArtal and Tardós with both monocular and stereo/RGB-D functionality. This algorithm has gained immense popularity due to its well-documented, usable source code, as well as its excellent speed and accuracy. This is one of the common benchmarks when comparing SLAM algorithms. In addition, lots of researches have noted the limitations of feature points. Works such as [7]–[10] have used line feature to improve system robustness because line provides significantly more geometrical structure information on the environment.

Not only applied to Visual SLAM, feature extraction has been a fundamental topic in computer vision. With the gradual maturity of the Visual SLAM framework, lots of works focus on the research of feature extraction. Ganti [11] considers that when selecting reference points for Visual SLAM, these points should meet several criteria. They should be: 1) Viewpoint invariant, 2) Scale-invariant, 3) Rotation invariant, 4) Illumination invariant, 5) Season invariant and 6) Static. Traditional feature detectors and descriptors, such as SIFT [12], SURF [13], or ORB [14] aim to tackle the first 3 criteria. However, many scholars have gradually realized the limitations of the design of traditional feature algorithms. Zhang and Vela *et al.* [15] consider that not all measured features in SLAM contribute to accurate localization during the estimation process. So, they describe a method for selecting a subset of features that are of high utility for localization in the SLAM estimation process. Similarly, Zhang *et al.* [16] propose an extra filtering strategy on current common features to efficiently reduce drift. They select features with the most contribution according to both spatial and temporal factors to reduce computation during bundle adjustment without losing accuracy. Belter *et al.* [17] investigate the influence of the uncertainty models of point features on the accuracy of the estimated trajectory and map in more detail and propose mathematical uncertainty models for point features in RGB-D SLAM. Unlike research on filtering strategies, Zhang *et al.* [18] carry out researches on randomized local binary features and propose using more general randomized intensity difference sampling operator to construct binary feature space for keypoints recognition. Yu *et al.* [19] propose a novel perspective invariant feature transform (PIFT) for RGBD images. In the work, they also point out that there are ''fake keypoints'' in a single 2D image, which cannot be distinguished or removed because of the lack of spatial information.

Similarly, our work also focuses on the processing of feature points in Visual SLAM to improve the results. Most of the published feature-based SLAM research still neglect the impact of background environment transformation on features. Typically, two steps can be distinguished in the utilization of visual features: The first step is the detection of interest points which should be detected at different distances and viewing angles. The second step is the feature descriptors of the selected point which usually is a feature vector computed from the surrounding information. The descriptor is used to solve the data association problem: when the robot observes a landmark in the environment, it must decide whether the observation corresponds to a previously seen landmark or to a new one. Such methods are based on a hypothesis that surrounding information of the point can stably represent the characteristics of the point [20]. However, the descriptor may fail to describe the feature at the edge of an object when the viewpoint changes with significant background transformation because the local information of the point will also be affected. In this case, the same point may fail to be matched for its quite different descriptor and the different point also may be mismatched for similar background appearance.

Unlike [19], which relies on the depth to filter fake keypoints, we believe that the unstable features are caused by the edge of the object. Recognition of objects in images has been a classic problem in computer vision. In recent years, the resurrection of deep learning [21] has had a major impact on the current state-of-the-art in machine learning and computer vision. The lightweight neural network also significantly reduces the extra computing resources required by the SLAM system to introduce semantic information. Bowman *et al.* [22] formulate their SLAM problem to include inertial, geometric, and semantic constraints into a joint optimization framework. This efficient combination method makes the SLAM system run in real time. For quite some time, the impact of the dynamic targets on the scene is a key issue in SLAM. With the help of deep learning, some works [23], [24] can eliminate this dynamic impact with semantic information. An *et al.* [5] propose a VO (Visual Odometry) pipeline which incorporates aspects of both indirect and direct SLAM methods as well as semantic information to reduce the effect of dynamic objects in the scene on the SLAM solution. Additionally, further combination works like [5] use quadrics as 3D landmark representations. These landmarks can be directly constrained via a novel geometric error formulation. In order to enhance robot's autonomy and robustness, facilitate more complex tasks, move from path-planning to task-planning, and enable advanced human-robot interaction, the importance of works of semantics of environments have been recognized. Based on the work [26], we continue to introduces semantics to solve the aforementioned problem of features located at the edges of objects and propose an effective edge-feature processing method for this. Inspired by Occam's Razor [27], we call our method Edge-Feature Razor, or EF-Razor for short.

EF-Razor uses the location of objects to distinguish the features belong to the edge regions. As shown in Figure 1, the features in the image can be divided into three categories: features at the edge of objects, features at the inside of objects

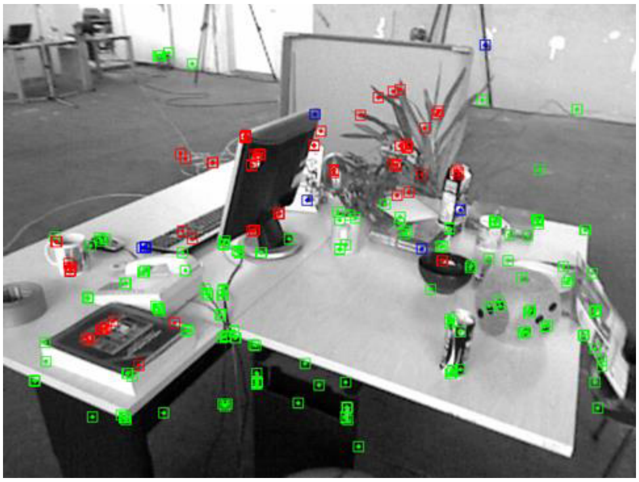


FIGURE 1. The features in the image can be divided into three categories: the edge region of objects (blue), the inside region of objects (red) and "no object" (green).

and features at the location which does not belong to any object. In this paper, the last type of feature is called "no object". The contributions of this paper are as follow:

1) We emphasize that the feature points in the different locations have different effects on the SLAM system and verified this view through statistics.

2) EF-Razor is proposed to use semantic information offered by the real-time object detection method YOLO [28] to process the duality of edge-features.

Through the experiments performed on the TUM RGB-D Dataset [29], EF-Razor has been proven to be effective in improving the positioning accuracy of the system.

II. THE PROPOSED METHOD

The foundation of EF-Razor is the observation and recognition of features. We use the phenomenon of descriptor failure under the change of perspective as an example to illustrate the viewpoint-dependent problem, and attribute the more fundamental reason to the lack of 3D spatial information. On this basis, EF-Razor eliminates the interference caused by such problems for SLAM systems.

A. THE EDGE-FEATURES ISSUE

Features usually appear as corners and can be numerously extracted in an image. Two steps can be distinguished in typical approaches for handcrafted features such as SIFT [12] and SURF [13]: 1. Selection of suitable points (e.g. points with large intensity variation) in the image; 2. Recording some properties (e.g. intensity gradient direction) of the neighborhood of the selected point as a matching basis. The pixel patches at the edge of the object are usually considered as the region of interest in the image by various algorithms because of the irregular shapes of many objects. In addition, the edge of an object and its background are often clearly distinguished, which means that the edge region has significant intensity variation.

Based on the understanding of features, we are aware that not all features in Visual SLAM contribute to accurate localization during the estimation process, especially edge-features. Although edge-features may be detected easily at different distances, they are sensitive to the perspective change because their surrounding pixel patches could be changed with different viewing angles. The descriptor of the handcrafted feature is calculated from the neighborhood according to a certain rule. Therefore, it will fail under the change of perspective with background transformation. Moreover, a similar background can also cause mismatched between two different key-points. An illustration of the above situation is illustrated in Figure 2. The actual matching point of a point $\mathrm{P_a}$ on the left is $\mathrm{P_b}$ in the right picture, but the matching point calculated by the descriptor of the ORB [14] is $\mathrm{P_c}$.

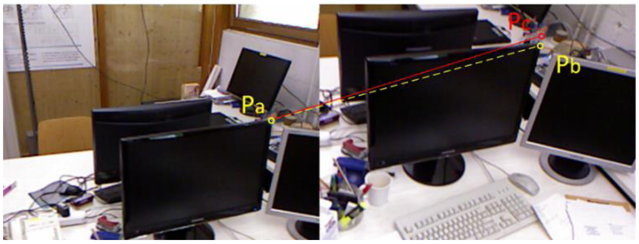


FIGURE 2. An illustration of mismatched edge-features. $\mathrm{P_a}$ is a key-point at the edge of a monitor and its actual matching point should be $\mathrm{P_b}$. $\mathrm{P_c}$ is the matching point calculated by the ORB [14].

In the application scenario of SLAM, an environment where objects overlap like Figure 2 is very common. A brief description used for SLAM should make it easy to understand some of the mechanisms that are responsible for the spatial uncertainty of the features. However, 2D image information is not sufficient to express defects in 3D space. The edge-feature patches usually are unstable because they are generated by the projection of two unconnected objects onto the 2D space. On the other hand, there are also features which are stable in 3D space among edge-features. They undoubtedly are contributive. In general, the edge-features issue can be attributed to their duality.

A part of unstable or similar feature can be removed by RANSAC [30] and other feature matching algorithms in Visual SLAM, considering the geometric consistence with respect to transformation. As a common tool, the RANSAC-based method is often used to reject outliers. Given an expected rate of success P, the necessary iteration times N could be computed by the number of data points S and outliers rate ε:

$$N = \log(1-P)/\log(1-(1-\varepsilon)^S) \tag{1}$$

With higher ε, N could reach thousands of times in many cases. In addition, the RANSAC-based method in SLAM assumes that the noise samples are far less than the correct samples. However, these edge-features can hardly be filtered out by RANSAC when the change of the views is not large enough. Therefore, if the edge-features are not effectively processed, the SLAM system may face the problems of excessive algorithm iterations and weakened algorithm assumption, which is also a value of our method. In EF-Razor, we are focused on effectively processing the features at the edges of an object in the Visual SLAM system.

B. FEATURES CLASSIFICATION

Classic corner detection can effectively detect the corners on the texture, but these tools cannot distinguish whether the corners are of the internal texture of the object or the external contour of the object. Therefore, we need to use YOLOv3 [28] that can accurately identify the position of the object. The basis of EF-Razor is using YOLOv3 to get the object position in each image. Meanwhile, these images are also processed by Visual SLAM. Then features are divided into different categories based on their positional relationship with the bounding box of objects. The detection performance of the deep object detector is mainly affected by the misjudgment of the object category. However, EF-Razor only needs the position information of bounding box, so it will not be affected by the object class of bounding box detected by mistake. This reduces EF-Razor's dependence on object detection accuracy. The four parameters we need for the bounding box are its center position on the image: x_0 and y_0; its length and width on the image: h and w. Based on the coordinates of the features on the pixel plane and the parameters of the bounding box, features can be classified while extracting them from the image. Considering that there may be multiple bounding boxes per frame, the feature, with a pixel coordinate of (x, y), needs to be compared to each bounding box. The detailed procedure is described in Algorithm 1.

Algorithm 1 Judging Feature Point Category

```
Input: feature pixel coordinates: $(x, y)$
Input: boudingbox center coordinates: $(x_0, y_0)$;
       boudingbox height: $hei$; boudingbox width: $wid$;
       inside factor: $in$; outside factor: $out$;
Input: class{$edge, inside, no\_object$}
Output: class{$edge, inside, no\_object$}
function EXTRACTLABEL (($x, y$), $hei$, $wid$, $class$)
    $x_{dis} \leftarrow |x - x_0|$
    $y_{dis} \leftarrow |y - y_0|$
    $class \leftarrow no\_object$
    if $x_{dis} > in \times hei$ and $x_{dis} < out \times hei$ and
      $y_{dis} > in \times wid$ and $y_{dis} < out \times wid$ then
      $class \leftarrow edge$
    end if
    if $x_{dis} < in \times hei$ and $y_{dis} < in \times wid$ then
      if $class$ is not $edge$ then
          $class \leftarrow inside$
      end if
    end if
end function
```

In the feature point category, the edge has the highest priority. Once a feature is judged to be an edge type, the remaining bounding boxes will no longer be considered. In contrary, the "no object" category has the lowest priority. A feature is judged as "no object" only if it is not within the scope of any bounding box. The specific classification range of the feature depends on the coefficients *in* and *out*. In this paper, *in* is 0.75 and *out* is 1.1. Limited by the rough object range of the object detection tool, Algorithm 1 does not accurately distinguish all edge features. But this is enough for EF-Razor to work.

C. FILTERING UNSTABLE EDGE-FEATURES

According to the above process, we can get the classification results of the features. This is also the basis of EF-Razor's role in SLAM. Unstable edge-features will undoubtedly damage the calculation results of SLAM. On the contrary, stable edge-features can provide stable descriptors for a long time, which is conducive to SLAM pose calculation. Based on the above facts, EF-Razor first filter out unstable edge-features, and then the stable edge-features can provide their best values in SLAM system. As known, the one-to-one match between the image keypoints is a key process of feature-based SLAM. This almost determines the initial pose resolution quality of the entire Visual SLAM front-end pipeline. Although there are numerous different methods for feature matching, the common ground of these methods is that their fundamental judgment is based on the approximation of feature descriptors. We first process the edge-feature points in the feature matching step. After classifying features, EF-Razor defaults that all edge-features are unstable. When performing matching, we implement stricter criteria for the keypoint pairs that match the edge-features or belong to edge-features in the current frame. In this way, when the object is occasionally lost in consecutive frames, EF-Razor can also process matching pairs containing edge-features. The keypoints pairs with large changes in the pixel patch around the feature are more likely to be mismatches caused by background changes. We believe that a pair of key points with closer descriptors when matching between frames can provide better results for feature-based Visual SLAM calculations. Reducing the maximum threshold of the bias of the descriptor in match will effectively eliminate the unstable edge-features.

D. VALUING CONTRIBUTIVE EDGE-FEATURES

After removing the unstable part of edge-features, the rest becomes positive in the SLAM system. EF-Razor take advantage of them to maximize the information gain in the estimation. By matching the feature points and performing a preliminary pose solution, the system can obtain the pose $(\boldsymbol{R_i}, \boldsymbol{t_i})$ of the i-th frame. $\boldsymbol{R_i}$ is called Rotation Matrix and $\boldsymbol{t_i}$ is called Translation Vector. With the initial pose, the system can calculate the distance between the projection of the 3D keypoints on the image and the detected keypoint. With this distance as the optimization objective, accurate pose can be solved by BA. The relationship between pixel position and spatial position can be represented as:

$$u_{ij} = \pi(R_i^T(P_j - t_i)) \tag{2}$$

where π represents the projection of 3D keypoints to the pixel plane and $\boldsymbol{P_j}$ is a certain point in 3D space. u_ij is the pixel coordinate of the $\boldsymbol{P_j}$ projection, which is the pixel coordinate of the camera viewing the 3D space point $\boldsymbol{P_j}$ at the pose $(\boldsymbol{R_i}, \boldsymbol{t_i})$. Generally due to the camera pose and the noise of the observation point, there is an error in the above equation, which is called the reprojection residuals. Summing the reprojection residuals over a series of reference camera frames can be constructed as a least squares problem:

$$f(R, t, P) = arg \min_{R,T,P} (e_{ij}^T \cdot H \cdot e_{ij}) \tag{3}$$

where $\boldsymbol{e_{ij}}$ represents reprojection residuals:

$$e_{ij} = u_{ij} - \pi(R_i^T(P_j - t_i)) \tag{4}$$

BA solves the least squares to adjust the coordinates of the 3D space point $\boldsymbol{P}$ and pose $(\boldsymbol{R_i}, \boldsymbol{t_i})$. $\boldsymbol{H}$ is the information matrix, which can be computed by inverting the covariance matrix of the measurement. In the feature-based Visual SLAM, the importance of constraints of feature positions is defined by their information matrices. For example, in ORB-SLAM2 [2], the pose graph-optimized information matrix is generally related to the scale information corresponding to the keypoints.

As described above, after processing the feature points during the matching step, the edge-features optimized by BA will not be affected by the change of viewing angle. In other words, at this time, the system can value the stability of such features in the pose solution. Therefore, EF-Razor can set H in BA optimization to:

$$H = H_0 * (1 + edge\left(u_{ij}\right) * val) \tag{5}$$

where $\boldsymbol{H_0}$ represents the original information matrix in SLAM system. *edge*(u_{ij}) is 1 when u_{ij} is an edge-feature, otherwise 0. *val* indicates the degree of importance that system attaches to edge-features in BA optimization.

In summary, EF-Razor improves the positioning accuracy of the feature-based SLAM system and the result is confirmed in the next section.

III. EXPERIMENT

Several experiments are conducted to evaluate our EF-Razor. The dataset for the experiments is the TUM RGB-D Dataset [29], which is a publicly available dataset with ground truth. The RGB-D images in the dataset, collected in 640 × 480 pixels, are captured by a hand-held Kinect in indoor environments. Its time-synchronized ground truth is obtained by a motion capture system. Our experiments are performed on a servicer with Intel(R) Xeon(R) CPU E5-2630v3@ 2.4GHz and GPU is NVIDIA TITAN X Pascal. RAM is 64G and the graphics card RAM size is 12G.

A. STATISTICAL CHARACTERISTICS OF DIFFERENT CATEGORIES OF FEATURES IN THE SLAM WORKFLOW

First, we make a statistic on the contribution of three categories of feature points in feature-based SLAM. For the keypoints corresponding to the matched pairs with large errors in the graph optimization, BA judges them as outliers. In contrast, the remaining points are inliers. Inliers usually mean points that match correctly. Therefore, in a way, the proportion of inliers reflects the stability of a class of keypoints. In order to observe the influences of different types of features, inlier rate of a certain features can be used as the objective function to evaluate the contribution of different types of features. We selected 7 commonly used sequences in the TUM RGB-D Dataset and recorded the inlier rate of the three categories ("no object", edge, and inside) of feature in the process of tracking the local map for each frame. Specifically, this part of the experiment chose ORB-SLAM2 as the benchmarks.

Suppose the number of frames of each sequence is n; the number of inliers for the *i*-th frame is $\mathrm{n_i}$; and, the total number of key-points for the *i*-th frame is $\mathrm{m_i}$; The mean and the standard deviation (SD) of the inlier rate of each sequence are shown in Table 1.

TABLE 1. Statistical characteristics of different categories of features in the ORB-SLAM2 workflow.

Sequence		No Object	Edge-feature	Inside-feature
fr1/desk	Mean	0.47194	0.461007	**0.477043**
	SD	**0.11431**	0.232226	0.157721
fr1/desk2	Mean	0.418281	0.4115	**0.428164**
	SD	**0.125457**	0.221822	0.171498
fr1/room	Mean	**0.514002**	0.500751	0.509472
	SD	**0.128074**	0.246206	0.20217
fr2/desk	Mean	**0.552101**	0.539501	0.508814
	SD	**0.0842349**	0.196971	0.116207
fr2/xyz	Mean	0.639369	**0.710475**	0.645084
	SD	**0.0541565**	0.117175	0.087059
fr3/ long_ office_household	Mean	**0.553693**	0.5082	0.536428
	SD	**0.0866924**	0.200497	0.115828
fr3/ nostructure_ texture_near_withloop	Mean	0.688364	**0.70582**	0.70404
	SD	0.0839616	0.215398	**0.079537**

As can be seen from Table 1, in most of the sequences, the statistical characteristics of the "no object" category are the best. In contrast, the "edge- feature" is the worst. Specifically, in the two sequences "fr2/xyz" and "fr3/nostructure_texture_near_withloop", the "edge" is the best result of the three categories in terms of the mean of inlier rate. In the above two sequences, the camera moves perpendicular to imaging plane, whereas in the rest of the sequences, the camera rotates horizontally over a wide range. This result is consistent with our analysis of the edge-feature. When the viewing angle changes greatly, the descriptors of edge-features are prone to failure due to the unstable

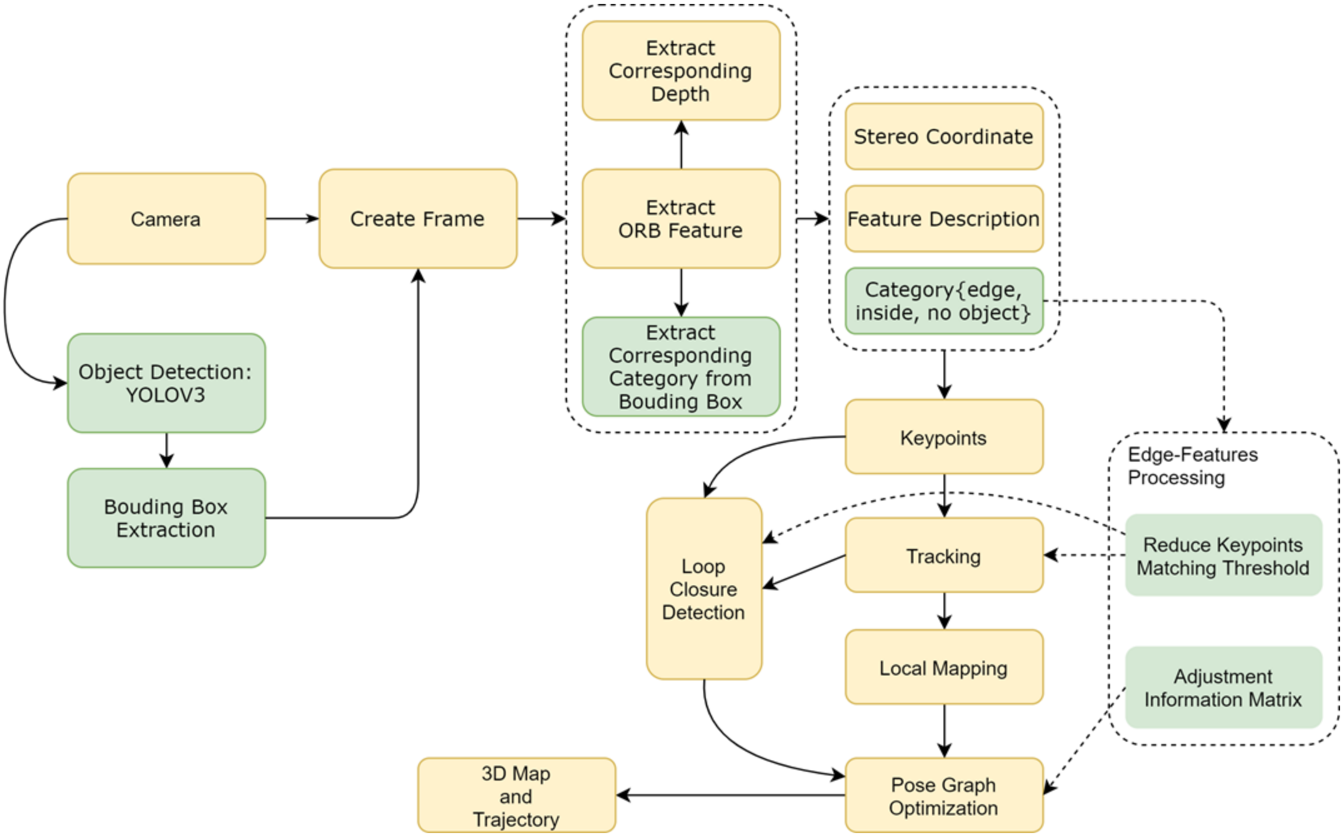


FIGURE 3. Overview of ORB-SLAM2 with EF-Razor added. (The yellow boxes are original ORB-SLAM2 [2] system module and the green boxes are ours).

connection state change in the image. On the contrary, when this "connected" state is maintained, the edge-features are quite contributive for the SLAM system. This also shows that the Algorithm 1 is effective as a whole.

For comparison, we set the maximum Hamming distance allowed by edge-features matching of the feature matching process to 10. Then, we recorded the inlier rate of the above seven sequences in the process of tracking the local map again. The results are shown in Table 2.

As listed in Table 2, after setting the edge-feature matching threshold to 10, the statistical characteristics of the edge-features in the above 7 sequences are all optimal. This proves that our processing method makes the edge-features that were originally restricted by the spatial structure instability become trustworthy. This also reflects that it is reasonable for us to value edge-features in the subsequent BA optimization.

TABLE 2. Statistical characteristics of different categories of features in the ORB-SLAM2 workflow (matching threshold is 10).

Sequence		No Object	Edge feature	Inside feature
fr1/desk	Mean	0.486837	**0.9799**	0.48754
	SD	0.1183	**0.10553**	0.15376
fr1/desk2	Mean	0.421562	**0.97114**	0.42504
	SD	0.118571	**0.11786**	0.166142
fr1/room	Mean	0.515166	**0.9814**	0.50922
	SD	0.124804	**0.08508**	0.206612
fr2/desk	Mean	0.554862	**0.99179**	0.506928
	SD	0.084247	**0.04141**	0.111681
fr2/xyz	Mean	0.623874	**0.99841**	0.639216
	SD	0.053733	**0.01959**	0.084118
fr3/ long_ office_household	Mean	0.552056	**0.99066**	0.537894
	SD	0.083802	**0.0452**	0.111365
fr3/ nostructure_	Mean	0.681853	**0.98563**	0.690921
texture_near_withloop	SD	0.082927	**0.07874**	0.094594

B. COMPARISONS WITH ORB-SLAM2

ORB-SLAM2 [2] is a feature-based SLAM system, which provides satisfactory results for static scenes of the TUM RGB-D Dataset [29]. We make some adjustments to the ORB-SLAM2 according to our EF-Razor. Compared with the original ORB-SLAM2 system, our changes based on EF-Razor are mainly on these modules: input, data association, ORB feature matcher and BA. EF-Razor first needs to integrate YOLOv3 [28] into ORB-SLAM2 to distinguish feature categories. This is reflected in the addition of an object detection module at the system input. Then, when creating a frame structure, EF-Razor associates the category information (edge, inside and "no object") from the object detection with the features. Finally, according to the categories, we make changes in feature matching and BA to improve

TABLE 3. ATE [m] of ORB-SLAM2 and ORB-SLAM2 with EF-Razor added in TUM RGBD dataset.

Sequence	Result refers to [2]	ORB-SLAM2		ORB-SLAM2+ EF-Razor		Improvement	
	RMSE	RMSE	SD	RMSE	SD	RMSE	SD
fr1/desk	0.016	0.0158	0.0014	**0.0145**	**0.000251**	**8.27%**	**82.07%**
fr2/desk2	0.022	0.0217	**0.0013**	**0.0214**	0.00202	**1.38%**	-55.38%
fr1/room	0.047	0.0474	0.0034	**0.0444**	**0.00228**	**6.33%**	**32.94%**
fr2/desk	0.009	0.0087	0.000823	**0.0079**	**0.000504**	**9.19%**	**38.76%**
fr2/xyz	0.004	0.0038	0.000261	**0.0035**	**0.000106**	**7.89%**	**59.39%**
fr3/long_office_household	0.01	0.0102	0.000927	**0.0088**	**0.00058**	**13.73%**	**37.43%**
fr3/nostructure_texture_near_withloop	0.019	0.0185	0.0027	**0.0178**	**0.0023**	**3.78%**	**14.81%**

our Visual SLAM. The framework of ORB-SLAM2 with EF-Razor added is shown in Figure 3. In this experiment, the val of the information matrix in the previous article is set to 0.15. The maximum Hamming distance allowed by the system's ORB descriptor matcher is 10.

ORB-SLAM2 is evaluated on the TUM RGBD benchmark before and after adding EF-Razor. To quantify the localization accuracy, the Absolute Trajectory Error (ATE) [29] that represents the global consistency is calculated with ground truth. Meanwhile, the standard deviation is applied for evaluating the stabilities of the approach. The quantitative results of the root mean squared error (RMSE) and the standard deviation (SD) are shown in Table 3. Meanwhile, the results in the paper [2] are also given in Table 3 as a reference.

As shown in Table 3, compared with original ORB-SLAM2, EF-Razor helps ORB-SLAM2 achieve better results in almost all sequences. We only get a poor standard deviation in sequence "fr1/desk2". The increased computation costs of EF-Razor lie primarily in the YOLOv3 modules. The proposed method can run at 16 Hz or greater.

In summary, our experiments indicate that edge-features have a negative impact on the SLAM system in most scenarios; EF-Razor can effectively improve the positioning accuracy of the SLAM system.

IV. CONCLUSION

This paper points out the edge-features issue in current common feature-based SLAM. If only considering the nature of the feature on the 2D image, the edge-features are easy to identify and suitable for long-term tracking. However, considering the change of the 3D structure, the edge-features are often in unstable regions. Aiming at the duality of edge-features, we have proposed EF-Razor to process them efficiently and this method can improve the SLAM system's ability to work in an environment where objects overlap. EF-Razor first needs object detection to distinguish edge-features. For defects of edge-features in 3D space, EF-Razor reduces such effects during the matching process. On the contrary, the edge-features that retain the advantages of the 2D image after the matching process will be more valued in EF-Razor, which is reflected in the optimization. In the experimental part, we verified the opinions of edge-features through statistical analysis. In addition, the experiments on public datasets show that EF-Razor is effective. We achieve an average 7% RMSE improvement on ATE.

The work in this paper makes use of the semantic information provided by object detection, more specifically the location information of the object. However, the concept of semantic information is very rich. In future work, the logical relationship between semantic information can effectively correct the misunderstanding of the environment by SLAM systems.

REFERENCES

[1] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[2] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[3] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016, doi: 10.1109/TRO.2016.2624754.

[4] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, Nov. 2007, pp. 1–10.

[5] L. An, X. Zhang, H. Gao, and Y. Liu, "Semantic segmentation–aided visual odometry for urban autonomous driving," *Int. J. Adv. Robot. Syst.*, vol. 14, no. 5, pp. 1–11, Sep. 2017, doi: 10.1177/1729881417735667.

[6] Q. Yu, J. Xiao, H. Lu, and Z. Zheng, "Hybrid-residual-based RGBD visual odometry," *IEEE Access*, vol. 6, pp. 28540–28551, 2018, doi: 10.1109/ACCESS.2018.2836928.

[7] X. Zuo, X. Xie, Y. Liu, and G. Huang, "Robust visual SLAM with point and line features," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Vancouver, BC, Canada, Sep. 2017, pp. 1775–1782.

[8] R. Guo, K. Peng, D. Zhou, and Y. Liu, "Robust visual compass using hybrid features for indoor environments," *Electronics*, vol. 8, no. 2, p. 220, Feb. 2019, doi: 10.3390/electronics8020220.

[9] P. Kim, B. Coltin, and H. J. Kim, "Low-drift visual odometry in structured environments by decoupling rotational and translational motion," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2018, pp. 7247–7253, doi: 10.1109/ICRA.2018.8463207.

[10] Y. He, J. Zhao, Y. Guo, W. He, and K. Yuan, "PL-VIO: Tightly-coupled monocular visual–inertial odometry using point and line features," *Sensors*, vol. 18, no. 4, p. 1159, Apr. 2018, doi: 10.3390/s18041159.

[11] P. Ganti, "Semantically informed visual odometry and mapping," M.S. thesis, Dept. Mech. Mechtron. Eng., Waterloo Univ., Waterloo, ON, Canada, 2018.

[12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[13] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[14] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.

[15] G. Zhang and P. A. Vela, "Good features to track for visual SLAM," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1373–1382, doi: 10.1109/CVPR.2015.7298743.

[16] H. Zhang, K. Hasith, and H. Wang, "A hybrid feature parametrization for improving stereo-SLAM consistency," in *Proc. 13th IEEE Int. Conf. Control Automat. (ICCA)*, Jul. 2017, pp. 1021–1026, doi: 10.1109/ICCA.2017.8003201.

[17] D. Belter, M. Nowicki, and P. Skrzypczyński, "Modeling spatial uncertainty of point features in feature-based RGB-D SLAM," *Mach. Vis. Appl.*, vol. 29, no. 5, pp. 827–844, Jul. 2018, doi: 10.1007/s00138-018-0936-9.

[18] J. Zhang, Z. Feng, J. Zhang, and G. Li, "An improved randomized local binary features for keypoints recognition," *Sensors*, vol. 18, no. 6, p. 1937, Jun. 2018, doi: 10.3390/s18061937.

[19] Q. Yu, J. Liang, J. Xiao, H. Lu, and Z. Zheng, "A novel perspective invariant feature transform for RGB-D images," *Comput. Vis. Image Understand.*, vol. 167, pp. 109–120, Feb. 2018, doi: 10.1016/j.cviu.2017.12.001.

[20] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: A survey," *Found. Trends Comput. Graph. Vis.*, vol. 3, no. 3, pp. 177–280, Jan. 2007, doi: 10.1561/0600000017.

[21] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[22] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic SLAM," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Singapore, May 2017, pp. 1722–1729, doi: 10.1109/ICRA.2017.7989203.

[23] A. Holliday and G. Dudek, "Scale-robust localization using general object landmarks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Madrid, Spain, Oct. 2018, pp. 1688–1694.

[24] P. Li, T. Qin, and S. Shen, "Stereo vision-based semantic 3D object and ego-motion tracking for autonomous driving," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 664–679.

[25] L. Nicholson, M. Milford, and N. Sunderhauf, "QuadricSLAM: Dual quadrics from object detections as landmarks in object-oriented SLAM," *IEEE Robot. Autom. Lett.*, vol. 4, no. 1, pp. 1–8, Jan. 2019, doi: 10.1109/LRA.2018.2866205.

[26] W. Liu, Y. Mo, and J. Jiao, "An efficient edge-feature constraint visual SLAM," in *Proc. Int. Conf. Artif. Intell., Inf. Process. Cloud Comput. (AIIPCC)*, Sanya, China, 2019, pp. 1–7, doi: 10.1145/3371425.3371455.

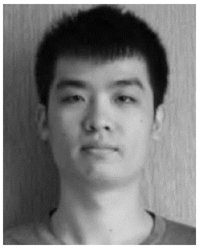
[27] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, "Occam's razor," *Inf. Process. Lett.*, vol. 24, no. 6, pp. 377–380, Apr. 1987, doi: 10.1016/0020-0190(87)90114-1.

[28] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," Apr. 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

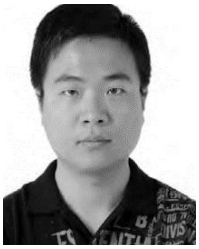
[29] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, Art. no. 573580.

[30] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

WEN LIU was born in 1967. She received the B.S. degree from the Xi'an University of Technology, in 1990, and the Ph.D. degree from the Beijing University of Posts and Telecommunications, in 2013. She is currently a Senior Engineer with the Wireless Network Positioning and Communication Fusion Research Center, School of Electronics Engineering, Beijing University of Posts and Telecommunications. Her main research areas are indoor positioning, wireless sensor networks, and multi-integrated positioning.



YAOKAI MO received the B.Eng. degree in electronic science and technology from the Beijing University of Posts and Telecommunications, Beijing, China, in 2017, where he is currently pursuing the master's degree with the Laboratory of Intelligent Communication, Navigation, and Micro/Nano-Systems. His research interests include computer vision, instance segmentation, and deep learning in SLAM.



JICHAO JIAO was born in Jining, China, in 1983. He received the B.S. degree in electronic information engineering from Yanshan University, Qinhuangdao, China, in 2007, and the Ph.D. degree in signal and information processing from the Beijing Institute of Technology, in 2013. Since 2013, he has been with the Beijing University of Posts and Telecommunications, where he has been an Associate Professor, since 2017. He is a Senior Member of the China Institute of Communications. His research interests include computer vision, vision-based navigation and indoor positioning, 3D semantic segmentation, and deep learning in SLAM.

ZHONGLIANG DENG (Senior Member, IEEE) received the M.Sc. degree in manufacturing engineering from Beihang University and the Ph.D. degree in mechanical manufacturing from Tsinghua University, China. He is currently a Professor and a Doctoral Supervisor with the School of Electronics Engineering, Beijing University of Posts and Telecommunications (BUPT). He is also the Director of research with the Laboratory of Intelligent Communication, Navigation, and Micro/Nano-Systems (ICNMNS). He is also the Executive Vice President of the BUPT. His research interests include indoor and outdoor seamless positioning, GNSS, satellite communications, MEMS, and multimedia.