A New Real-Time Visual SLAM Algorithm Based on the Improved FAST Features

Liang Wang[1], Rong Liu[2], Chao Liang[1], and Fuqing Duan[3]

[1] College of Electronic Information and Control Engineering,
Beijing University of Technology, Beijing 100124, China
[2] Base Department, Beijing Institute of Fashion Technology, Beijing 100029, China
[3] College of Information Science and Technology,
Beijing Normal University, Beijing 100875, China
wangliang@bjut.edu.cn

Abstract. The visual SLAM is less dependent on hardware, so it attracts growing interests. However, the visual SLAM, especially the Extend Kalman Filter-based monocular SLAM is computational expensive, and is hard to fulfill real-time process. In this paper, we propose an algorithm, which uses the binary robust independent elementary Features descriptor to describe the features from accelerated segment test feature aiming at improving feature points extraction and matching, and combines with the 1-point random sample consensus strategy to speedup the EKF-based visual SLAM. The proposed algorithm can improve the robustness of the EKF-based visual SLAM and make it operate in real-time. Experimental results validate the proposed algorithm.

1 Introduction

The simultaneous localization and mapping (SLAM) is a problem that if it is possible, for a robot or autonomous vehicle to be placed at an unknown environment, to build a map of the environment and at the same time use this map to determine its location [1–3]. According the type of sensors, the SLAM techniques in literatures can be roughly categorized into laser-based, sonar-based, Global Positioning System (GPS)-based and vision-based techniques. In comparison with other sensors, the visual sensors are passive, which have high resolution, long range and low dependence on hardware, and can provide both depth and appearance information of the environment with one (image) measurement. So there is a growing interest in vision-based (or visual) SLAM.

Generally, there are two types of vision-based SLAM systems: the monocular SLAM and the binocular SLAM. In comparison with the latter, the former has less dependence on the hardware and more flexibility. Then the monocular SLAM is the focus of the research community. For simplicity, objects in the scene are usually assumed to be static. The monocular SLAM is performed by obtaining the depth information as follows. The camera captured images of features in the scene during its moving. Then with the knowledge of computer vision, the depth information of each feature observed in both current and last frame can be

M. Kurosu (Ed.): Human-Computer Interaction, Part V, HCII 2013, LNCS 8008, pp. 206–215, 2013.

computed. With the reconstructed 3D information of features, the environment map consisting of sparse features is building, and the localization information of the camera is also obtained.

Currently, most of the monocular SLAM algorithms are based on image point features. One representative work is the Parallel Tracking and Mapping (PTAM) algorithm proposed by G.Klein and D.Murray [4]. The PTAM algorithm splits the SLAM into two separate tasks: tracking and mapping. These tasks are processed in parallel threads: one thread deals with the task of robustly tracking camera motion, while the other produces a 3D map of feature points from previously observed images. With robust and sparse feature points, the tracking can robustly retrieve the camera trajectory and the asynchronously mapping can provide a dense map. This algorithm has a high precision. However its computational cost dramatically increases with the increase of the number of feature points. So it is limited to the indoor scene and applications with little movement in a small region, and not suitable for general applications. The other representative work is due to Davison et al. [1], which performs a recursive updating of scene structure and camera motion estimates using an Extended Kalman Filter (EKF). It has better environment adaptability and can fulfill task in large areas such as several street blocks, one city and so on [5]. However, the high computational cost is the bottleneck of the EKF-based SLAM. The main reason for its high computational cost comes from that most image point feature extraction methods used in the EKF-based SLAM cannot trade off the speed of extraction and the reliability of matching.

In this paper, we propose to use the Binary Robust Independent Elementary Features(BRIEF) descriptor to describe the Features from Accelerated Segment Test (FAST) aiming at improving feature points extraction and matching, and combine it with the 1-point RANdom Sample Consensus (RANSAC) strategy to improve the EKF-based visual SLAM. The BRIEF descriptor uses a binary sequence to describe the extracted feature point in the extraction process. Then in the matching process, the Hamming distance is calculated to perform features matching, which is just the XOR calculation of two sequences in bitwise. It can better trade off the speed of feature extraction and the reliability of feature matching. The 1-point RANSAC can reduce the number of the correction in the updating step of the EKF, i.e. reduce the dimension of the state, to improve the calculation speed. By incorporating the BRIEF and 1-point RANSAC strategy into the EKF-SLAM, we improve the robustness of the EKF-based SLAM, which is also satisfying real-time demands. Experimental results validate the proposed algorithms.

The paper is organized as follows: In Section 2 the computational framework of the EKF-based visual SLAM is briefly discussed. Then we show how to incorporate the improved FAST features and 1-point RANSAC strategy into the EKF-SLAM in Section 3 and 4 respectively. In Section 5 some experimental results are presented. Finally the paper is concluded in Section 6.

2 The EKF-Based Visual SLAM

The EKF is the extension of Kalman filter. It can be used to handle the nonlinearity in transfer matrix and measurement function, and the nonlinear noise in the linear system. It is suitable to describe most of physical systems. The mathematical model of the EKF-based SLAM can be summarized as follows [2].

2.1 The State Description of the Visual SLAM System

In EKF-based monocular SLAM, the state vector of a camera, $\mathbf{x}_v$, has 13 elements, which consists of a metric 3D position vector $\mathbf{p}^W$, orientation quaternion $\mathbf{q}^{CW}$, velocity vector $\mathbf{v}^W$, and angular velocity vector ω^C relative to a fixed world frame W and camera frame C.

$$\mathbf{x}_v = \begin{bmatrix} \mathbf{p}^W \\ \mathbf{q}^{CW} \\ \mathbf{v}^W \\ \omega^C \end{bmatrix} \tag{1}$$

The state of the visual SLAM system can be described by a state vector $\hat{\mathbf{x}}$ and its covariance matrix P. State vector $\hat{\mathbf{x}}$ comprises the stacked state estimates of the camera $\hat{\mathbf{x}}_v$ and feature points $\hat{\mathbf{y}}_i$.Covariance matrix P is a square matrix which can be partitioned into submatrix elements as follows.

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_v \\ \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2 \\ \vdots \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} \mathbf{P}_{xx} & \mathbf{P}_{xy1} & \mathbf{P}_{xy1} & \cdots \\ \mathbf{P}_{y1x} & \mathbf{P}_{y1y1} & \mathbf{P}_{y1y2} & \cdots \\ \mathbf{P}_{y2x} & \mathbf{P}_{y2y1} & \mathbf{P}_{y2y2} & \cdots \\ \vdots & \vdots & \vdots & \end{bmatrix} \tag{2}$$

2.2 The Motion Model of Camera

The "constant velocity, constant angular velocity model" is used to describe the motion of camera [2]. It is assumed that in each time step acceleration $\mathbf{a}^W$ and angular acceleration α^W observe the zero mean and Gaussian distribution. Then the velocity and angular velocity follow

$$\mathbf{n} = \begin{bmatrix} \mathbf{V}^W \\ \Omega^C \end{bmatrix} = \begin{bmatrix} \mathbf{a}^W \Delta t \\ \Omega^C \Delta t \end{bmatrix} \tag{3}$$

Assuming that the covariance matrix of the vector $\mathbf{n}$ is diagonal, the state update procedure is

$$\mathbf{f}_v = \begin{bmatrix} \mathbf{p}^W + (\mathbf{V}^W + \mathbf{v}^W)\Delta t \\ \mathbf{q}^{WC} \times \mathbf{q}((\mathbf{V}^W + \mathbf{v}^W)\Delta t) \\ \mathbf{v}^W + \mathbf{V}^W \\ \omega^W + \Omega^W \end{bmatrix} \tag{4}$$

where $\mathbf{q}((\mathbf{V}^W + \mathbf{v}^W)\Delta t)$ the quaternion corresponding to the angle-axis rotation vector $(\mathbf{V}^W + \mathbf{v}^W)\Delta t$.

In the EKF, the state uncertainty (covariance matrix) Q_v of the new state estimate $\mathbf{f}_v(\mathbf{x}_v, \mathbf{u})$ for the camera can be computed via the Jacobian matrix:

$$\mathbf{Q}_v = \frac{\partial \mathbf{f}_v}{\partial n} \mathbf{p}_n \frac{\partial \mathbf{f}_v}{\partial n}^T \tag{5}$$

where $\mathbf{p}_n$ is the covariance matrix of vector $\mathbf{n}$.

2.3 The Measurement Model of Camera

The measurement model uses the observed feature points in the scene and position of camera to predict the image position of each feature point before deciding which is to be measured. Then actively measure selected feature points to update the EKF. It firstly transforms feature point from the world coordinate system to the camera coordinate system

$$\mathbf{h}_i^C = \begin{bmatrix} \mathbf{x}_i^C \\ \mathbf{y}_i^C \\ \mathbf{z}_i^C \end{bmatrix} = \mathbf{R}^{CW}(\mathbf{y}_i^W - \mathbf{p}^W) \tag{6}$$

Then the image coordinates can be found using the standard pinhole camera model [6]

$$\mathbf{h}_i = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_0 - f_u h_x / h_z \\ v_0 - f_v h_x / h_z \end{bmatrix} \tag{7}$$

where f_u, f_v is the focal length along u and v axis respectively, (u_0, v_0) is the principal point.

Finally, the predicted image coordinates of the feature can be determined with the distortion parameter K_1

$$\begin{cases} u_d = u_0 + \frac{u-u_0}{\sqrt{1+2K_1[(u-u_0)^2+(v-v_0)^2]}} \\ v_d = v_0 + \frac{v-v_0}{\sqrt{1+2K_1[(u-u_0)^2+(v-v_0)^2]}} \end{cases} . \tag{8}$$

2.4 The Iteration Process of the EKF

The iteration process of the EKF can be divided into two steps: prediction and updating [2]. In the prediction step, system can predict the state of the next step. Then in the updating step, the predicted parameters can be rectified with current measurements. The processes can be express as follows

$$\hat{\mathbf{x}}_{new} = \hat{\mathbf{x}}_{old} + W(\mathbf{z}_i - \mathbf{h}_i) \tag{9}$$

$$\hat{\mathbf{p}}_{new} = \hat{\mathbf{p}}_{old} - W\mathbf{S}W^T \tag{10}$$

where $\mathbf{z}_i$ and $\mathbf{h}_i$ denotes the measured and predicted value of the feature point respectively, W denotes the Kalman gain,

$$W = p\frac{\partial \mathbf{h}_i}{\partial \mathbf{x}}^T S^{-1} = S^{-1} \begin{bmatrix} p_{xx} \\ p_{y1x} \\ p_{y2x} \\ \vdots \end{bmatrix} \frac{\partial \mathbf{h}_i}{\partial \mathbf{x}_v}^T + S^{-1} \begin{bmatrix} p_{xyi} \\ p_{y1yi} \\ p_{y2yi} \\ \vdots \end{bmatrix} \frac{\partial \mathbf{h}_i}{\partial \mathbf{x}_v}^T \tag{11}$$

where S denotes the uncertainty of features, the uncertainty of each feature can be expressed as

$$S_i = \frac{\partial u_{di}}{\partial \mathbf{x}_v} p_{xx} \frac{\partial u_{di}}{\partial \mathbf{x}_v}^T + \frac{\partial u_{di}}{\partial \mathbf{x}_v} p_{xyi} \frac{\partial u_{di}}{\partial \mathbf{x}_{yi}}^T + \frac{\partial u_{di}}{\partial \mathbf{x}_{yi}} p_{yix} \frac{\partial u_{di}}{\partial \mathbf{x}_v}^T + \frac{\partial u_{di}}{\partial \mathbf{x}_{yi}} p_{yiyi} \frac{\partial u_{di}}{\partial \mathbf{x}_{yi}}^T \quad (12)$$

where R is the noise matrix.

3 The Extraction and Matching of Point Feature

Most of the visual SLAM is based on point features, which takes image points corresponding to features in the scene as the landmark to construct 3D map and perform localization. So the quality of extraction and matching of feature points in images has great impact on the visual SLAM. Currently, the most popular feature point is the Scale-Invariant Feature Transform (SIFT) point [2]. The SIFT feature uses a vector with 128 elements to describe a feature point [8]. It is invariant to location, scale and rotation, and partially invariant to illumination changes. However the computational cost is too large to satisfy the real-time demands. The FAST [9] with BRIEF description [10] are used to perform feature extraction and matching.

3.1 Feature Extraction

The FAST point [9] is extracted from the scene to fasten the feature extraction. In real applications, each image is partitioned into several blocks and feature extraction is performed in each block. Firstly, randomly select one image coordinates. Take a 20×30 rectangular region whose center is the selected image coordinates. Then determine whether a feature point has been selected from this region. If there is no FAST point, perform FAST feature extraction and select the point with the highest precision as the FAST point of this region. Otherwise, re-select the image coordinates and repeat the former step until it reaches the threshold of the feature point number or the maximum iteration times. This method can reduce the computational cost and avoid selecting too dense features in some regions.

Once FAST points are extracted, the BRIEF descriptor [10] is used describe these feature points, which is a binary sequence. BRIEF is based on comparisons. Take a patch with a FAST point locating at the center of the patch. Then choose one point other than the center in this patch. Compare intensities of the selected point and the center. If the intensity of the selected point is larger than that of the center, assign the value '1', else '0'. Do that for each point in the patch except the center and end up with a string of boolean values. To improve the robustness, the image patch can be filtered by the Gaussian filter before comparison.

3.2 Feature Matching

Feature matching is performed by calculating the Hamming distance between BRIEF descriptors of two extracted FAST points. These two feature points come

form the current image and the next image respectively. The Hamming distance is determined by bit-wise XOR operation. Only the two feature points whose computed Hamming distance is less than the threshold, are considered as the corresponding point. Due to the existence of regions with similarity texture, false matches inevitably occur. Since the motion of camera is continuous, positions of the corresponding feature point in current and next image should not be far away from each other. A 20×20 square region is used to narrow the match region. If two matching feature points are covered by a 20×20 square, the match is considered as a correct one, otherwise it is regarded as a false match and deleted.

3.3 Feature Fusion

After feature matching, the EKF can be updated by exploiting coordinates of the predicted feature points and coordinates of matched feature points. Do not delete those feature points without matches in the next image immediately. This is because in the real application, due to the influence of light changes or motion blur, some feature points would not be extracted in some images, however would appear again in later images. So in this case, these feature points are first marked, and cannot be used to update the EKF. The predicted values instead of coordinates of matched feature points are used to update the EKF. Once one feature does not find correct match in continuous five frames, the feature point is considered as lost and deleted.

4 Incorporating the 1-Point RANSAC into the EKF-Based SLAM

The computational framework of the EKF is introduced in Section 2. It is obvious that the computational cost is very high, especially in the updating step whose computational complexity is $O(M * N^2)$, where N is the dimension of the EKF state, M is the number of feature points. This is one important reason that the existing algorithms do not satisfy the real-time demands. In this section, we incorporating the 1-Point RANSAC into the EKF-based SLAM to reduce the computational cost.

The 1-Point RANSAC algorithm is due to Civera et al. [7]. The main contribution is that it greatly reduce computational cost of the updating step of the EKF. This is obtained by introducing the RANSAC scheme. Then the iteration number of updating step and the dimension number of the state are greatly reduced. It has following two steps.

4.1 Inlier Processing

The input of the 1-Point RANSAC algorithm is measured feature points z. Firstly the probability of measured feature point $\eta(h_K(\hat{\mathbf{x}}_{K|K-1}), S_K)$ is computed. Those feature points whose probability are equal to or greater than 99% are considered as inliers. Then make random sample from obtained inliers by

the RANSAC scheme. In this process, to reduce the computational cost, only the state vector $\hat{\mathbf{x}}$, rather than the covariance matrix P, is updated. Finally, the number of iteration in updating step of the EKF is computed with

$$n_{hyp} = \frac{log(1-p)}{log(1-(1-\epsilon)^m)} \tag{13}$$

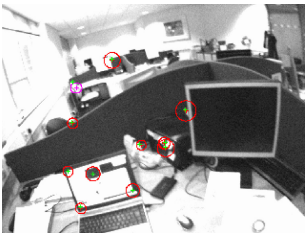
where p denotes the desired precision, ϵ is the inlier rate, $m = 1$.

4.2 Outlier Processing

The extracted feature points also contain the matched feature points with low precision and mismatched points besides inliers. The matched feature points with low precision can be divided into two categories: one class are feature points extracted in the first time whose low precision is due to the first extraction and limited number of iteration, the other are feature points near the camera. To improve the precision, we update each matched feature point with low precision and discard the mismatched feature points. First, use the state vector and covariance matrix obtained in inlier processing step to perform updating. Then validate all outlier. Only those feature points whose correlation rate is greater than 99% are regarded as inliers and the others are discarded.

(a) one frame of captured video

(b) feature points in one frame



(c) one frame with discarded feature point

(d) one frame with unmatched feature point

Fig. 1. Sample frames of video captured by the camera and feature points used to perform the proposed algorithm

5 Experiments

To validate the proposed algorithm, some experiments have been performed. One of them is reported here. The data are from the University of Zaragoza [7]. The proposed algorithm is developed with Visual C++ and some functions of the OpenCV. It runs on a notebook computer with a P4 2.4GHZ CPU.

Figure 1(a) shows the first frame of the video captured by the camera performed the monocular SLAM. Figure 1(b) shows one frame of the video, in which feature points are extracted and matched and the red ellipse is used to show that the corresponding feature point is a inlier with high innovation. The size of the ellipse presents the predicted feature point within this ellipse with a likelihood of 99%. With the running of the algorithm, the precision of prediction increases. In figure 1(c), we can see that the size of red ellipse are greatly reduced. In figure 1(d), there is a magenta ellipse besides red ellipses, which is used to express the feature point discarded by the 1-Point RANSAC due to large error. In figure 1(c), there is a blue ellipse which is used to express the mismatched feature point. In experiments, we do not remove the feature points corresponding to the magenta and blue ellipses immediately. This is because some feature points would not be extracted in some images due to light changes or motion blur, however they would appear again in later frames. Figure 2 shows the final trajectory of the camera. Figure 3 shows the process time of each frame, where the largest one is about 47.8ms, the least one is about 5.6ms and the average is about 23.53ms. For real applications with a camera of 30 frames per second, the proposed algorithm can satisfy the real-time demands.

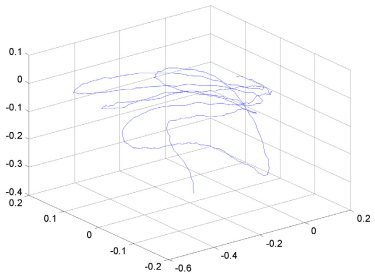


Fig. 2. Trajectory of the camera obtained by the proposed algorithm

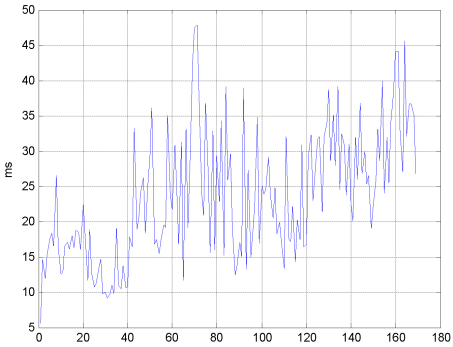


Fig. 3. Time consuming of the proposed algorithm

6 Conclusions

In this paper, we investigate the real-time monocular SLAM. A new algorithm is proposed to improve the EKF-based monocular SLAM. It incorporates the FAST feature detector with the BRIEF descriptor to extract and match image feature points and the 1-Point RANSAC strategy to improve the EKF-based SLAM. The proposed algorithm can improve the robustness of the EKF-based SLAM, which is also satisfying real-time demands. Experimental results validate the proposed algorithm.

Acknowledgment. This work was partially supported by the National Natural Science Foundation of China (Grant No.61101207).

References

1. Davison, A.J.: Real-time simultaneous localisation and mapping with a single camera. In: Proceedings of the International Conference on Computer Vision, Nice (2003)
2. Davison, A.J., Molton, N.D., Reid, I.D., Stasse, O.: MonoSLAM: Real-time single camera SLAM. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(6), 1052–1067 (2007)
3. Civera, J., Davison, A.J., Montiel, J.M.: Inverse depth parametrization for monocular SLAM. IEEE Transactions on Robotics 24(5), 932–945 (2008)
4. Klein, G., Murray, D.W.: Parallel tracking and mapping for small AR workspaces. In: Proceedings of the International Symposium on Mixed and Augmented Reality, pp. 1–10 (2007)
5. Strasdat, H., Montiel, J.M., Davison, A.J.: Visual SLAM: Why Filter? Image and Vision Computing 30(2), 65–77 (2012)
6. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision, 2nd edn. Cambridge University Press, Cambridge (2004)

7. Civera, J., Grasa, O.G., Davison, A.J., Montiel, J.M.: 1-Point RANSAC for EKF filtering: application to real-time structure from motion and visual odometry. Journal of Field Robotics 27(5), 609–631 (2010)
8. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Herbert Bay (1999)
9. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 430–443. Springer, Heidelberg (2006)
10. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: Binary Robust Independent Elementary Features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 778–792. Springer, Heidelberg (2010)