Feature Matching Based on Target Detection Optimization in Dynamic Scene

Shu WANG, Xin WU* and Shi-guang WEN
Northeastern University, Shenyang 110004, China
*Corresponding author

Keywords: Target detection, Dynamic scene, Feature matching, ORB_SLAM2.

Abstract. To improve the accuracy of ORB_SLAM2 pose estimation in dynamic environment, our paper proposes an effective method which uses the target detection algorithm to select the ROI of the key frame of the image, and then performs feature point matching. This method can be divided into three steps. Firstly, targets are detected and the ORB feature points are extracted from the input image, the corresponding pixel position coordinates of the object are eventually obtained in the scene; Secondly, feature point matching are performed on these ROI regions; Thirdly, the feature points with the relatively stable regions of interest are used to perform pose estimation. To verify the effectiveness of our method, the proposed method is tested on the public dataset. The experimental results show that the improved method can not only shorten the matching time, but also greatly improve the accuracy when compared with the traditional method. Thus, the original ORB_SLAM2 system can significantly improve the accuracy of pose estimation in dynamic environment.

Introduction

The process of acquiring external environmental information through sensors in an unknown environment, realizing pose estimation and incrementally constructing an environmental model, and then establishing its own global position is called Simultaneous Localization and Mapping [1]. The input to the SLAM algorithm is continuous sensor information. For example, in visual SLAM, the input information is a sequence of image frames continuously acquired by the camera-equipped robot in the environment. The output is the current position of the robot and the map point location of the environment in which the robot is located. Visual odometry (VO) is the most critical part of the whole SLAM system [2]. It is necessary to estimate the relative displacement of the two frames according to the position of the associated pixel between two adjacent frames. Therefore, the accuracy of the pose estimation depends on the accuracy of the coordinates of the associated feature points between two adjacent frames. If the feature point has a large matching error, it will cause an incorrect estimation of the pose estimation, resulting in the positioning failure.

Target detection [3] of an image refers to the process of separating the foreground [4] and background from the image and classifying the objects in the foreground. As an important branch of deep learning [5], image target detection has been widely used in automatic driving, scene recognition and medical image analysis in recent years. As shown in Fig. 1, Fig. 1(a) is the original image, and Fig. 1(b) is the result after object detection.

Current advances in deep learning provide a direction for solving this problem. The powerful feature learning ability of deep neural network has made significant progress in image recognition and target detection. In the game of image recognition dataset ImageNet, the deep learning model based on CNN [6] has achieved more than human performance. the deep learning method represented by RCNN [7] in target detection greatly improves the accuracy of target detection. Aiming at the problem that the matching speed and error of adjacent pixels in adjacent two frames are relatively large, this paper proposes a kind of visual SLAM combined with neural network based on target detection [8] algorithm to improve the efficiency and accuracy of feature point matching point pairs. Firstly, the ROI is extracted from the image by the target detection algorithm [9], and the target accurate region in the key frame of the image is obtained. Then the matching...
point pair is extracted by the ORB feature point matching algorithm. By using the algorithm of this paper, the task of feature matching can be realized better, and then the matching feature points can be more accurately used for pose estimation.

![Image](image.png)

(a)The original image               (b)Object detection result

Figure 1. Image target detection.

Traditional ORB_SLAM2

ORB_SLAM2 is a real-time SLAM system based on ORB feature points that can run on the CPU. It consists of three main threads: tracking thread, local mapping thread and loop detection thread, as shown in Fig. 2.

![Diagram](diagram.png)

Figure 2. ORB_SLAM2 system threads and modules.

The tracking thread, extract and match the ORB feature points, and then estimate the relative pose between the two frames by minimizing the re-projection error; the local mapping thread optimizes the pose of all frames in the local mapping by BA (Bundle Adjustment) The loop detection thread detects whether the entire map has closed loops through key frames, and continuously corrects the accumulated drift error by pose optimization. In SLAM, pose is the spatial position and attitude of the robot in the entire environment map. The spatial position is the \(xyz\) coordinate of the robot. The spatial attitude refers to the yaw angle of the robot in the positive direction (generally the direction of the camera) relative to the \(xyz\) direction. In ORB_SLAM2 [10], the pose of the robot is represented by a seven-ary number consisting of a translation vector plus a rotated quaternion, as shown in the following equation:
\[ T = [x, y, z, q_x, q_y, q_z] \]  

(1)

where the first ternary is the translation vector; the last quaternion is the quaternion representing the rotation.

The task of tracking threads is to calculate the pose of two adjacent frames according to the image change, that is, how much and how much the next frame is translated relative to the previous frame; then, the calculation results are given to the back end, and the back end will be the two or two frames. Accumulate and optimize the relative pose between the two; finally find the current pose of the robot to achieve positioning. The principle of solving the relative pose between two frames is shown in Fig. 3. The two frames before and after the camera are obtained. After the feature is extracted, the feature points are obtained and the feature points are obtained. It is assumed that according to the result of the feature matching, the pair of points that are closer to each other, that is, the projection of the same three-dimensional space point \( P \) on the two frames of images.

\[ p_1 = Kp, \quad p_2 = K(Tp) \]  

(2)

where \( K \) is the internal parameter matrix of the camera. When the camera is in different poses, the point \( P \) is transformed by the internal reference matrix to obtain different pixel coordinates, projection \( p_1, p_2 \) and \( T \) is the pose of \( I_2 \) relative to \( I_1 \). Assuming that multiple sets of point pairs can be matched between two frames, the equations can be constructed by these pairs of points to solve the relative pose. Specifically, it can be solved by solving the basic matrix and the homography matrix. However, the calculation of \( T \) must be effective under the condition that the spatial point \( P \) is stationary relative to the whole environment. If the point moves during the pose estimation, Eq. 2 is no longer valid. The error will follow. In the worst case, all pixels participating in the pose estimation perform the same motion with the camera, and the pose estimated by SLAM will always be zero.

SSD Object Detection

Reason for Choosing SSD Network

Object detection speed analysis. In order to achieve the requirement of constructing a semantic map in real time, the processing time and processing accuracy of the object detection are balanced. In order to measure the running time efficiency of the proposed algorithm, this paper tests the line efficiency of the object detection algorithm on the dataset to deal with the frame rate as the measurement standard, and with the high-precision Fast RCNN [11], Faster RCNN and other detection algorithms, and the real-time 100Hz DPM, 30Hz DPM detection algorithm is compared. The experimental platform is Ubuntu 18. The system environment of 04, the processor model is Intel i7-6700, the memory is 20 GB, and the graphics card model is GTX 1080Ti. Experimental knot As shown in Table 1.

It can be seen that the frame rate of Faster RCNN and Fast RCNN is much lower than that of SSD object detection [12] algorithm, and the real-time requirements cannot be met. Although the DPM algorithm runs the FPS index close to this article, the algorithm accuracy is poor. It can be
seen that the SSD algorithm has achieved a good balance in precision and time efficiency, and can achieve a good mAP on the basis of real-time operation, so this paper adopts the object detection algorithm of SSD.

### Table 1. Target detection speed comparison experiment results.

| Detection Algorithm | mAP/% | FPS |
|---------------------|-------|-----|
| 100hz DPM           | 16.0  | 100 |
| 30hz DPM            | 26.1  | 30  |
| Fast RCNN           | 67.32 | 0.5 |
| Faster RCNN         | 68.17 | 7   |
| SSD                 | 74.3  | 59  |

### Introduction to SSD Network

SSD is a convolution neural network structure with target detection. At the heart of the SSD approach is the use of small convolution filters to predict the category fraction and position offset of a set of default bounding boxes[14] fixed on a feature graph. In order to achieve high detection accuracy, we produce different scale predictions from different scales of feature diagrams, and clearly separate predictions by aspect ratio. In summary, these design features get simple end-to-end training and high precision, further increasing the trade-off between speed and precision, even if the input is relatively low resolution of the image. The SSD method is based on a feedforward convolution network [15], which produces a fixed-size set of bounding boxes and a fraction of the object categories in the box, followed by a non-maximized suppression step to produce final detection. The SSD core design concept is summarized as the following three points:

b) Detection by convolution: Different from Yolo’s last use of the fully connected layer, SSD directly uses convolution to extract the detection results from different feature maps. For a feature map having a shape of $m \times n \times p$, it is only necessary to use a relatively small convolution kernel of $3 \times 3 \times p$ to obtain a detection value.

c) Set the a priori box: The SSD draws on the idea of the anchor in the Faster R-CNN. Each unit sets a different a priori box of scale or aspect ratio. The predicted bounding boxes are based on these prior boxes, to a certain extent. Reduce the difficulty of training. In general, each unit will have multiple a priori boxes with differences in scale and aspect ratio.

Then, we add a secondary structure to the network, resulting in detection with the following main characteristics: Multi-scale feature graph detection: We add the convolution feature layer to the end of the truncated base network [16]. These layer sizes decrease gradually, and the predicted values of multiple scale detection are obtained. The convolution model detected is different for each feature layer, as shown in Fig. 4.

![Figure 4. Illustration of the SSD Architecture.](image)
Experiments
This experiment is to shoot books from two different angles, as shown in Fig. 5(a) and Fig. 5(b), and then through the SSD neural network to detect the target in the image, contrast the pattern image for feature matching and the selection of ROI after the feature matching time, as shown in Fig. 5(c) and Fig. 5(d).

Figure 5. The result of feature matching by the original image and the result of feature matching after selecting roi.
From the experimental results, we can see that the time of feature matching of the original image is 1.0351s, the time of feature matching after selecting the area of interest is 0.587384s, and the number of feature points matching the latter is much larger than that of the former. (SSD detection time is 100FPS/s, each frame image is 0.01s, relative to feature point matching time is negligible)

Summary

On the basis of ORB-SLAM2, this paper puts forward the complexity of the time of feature point matching by selecting the area of interest, and increases the accuracy and quantity of feature point matching, and proves the experimental results on the data set, which shows that The improved feature matching matching points can better make posture estimation and reduce relative drift, thus improving the accuracy of the whole system.

Acknowledgement

This work was supported by National Key R & D Program of China Number 2017YFB1301103, and the Fundamental Research Fund for the Central Universities of China N172604003, N172603001, and supported by Doctoral Foundation of Liaoning Science and Technology Department Number 20170520244, and the National Natural Science Foundation of China under Grant nos. 61701101, U1713216, 61803077, 61603080.

References

[1] Choset H, Nagatani K. Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization[J]. Robotics & Automation IEEE Transactions on, 2001, 17(2):125-137.
[2] Mur-Artal R., Montiel J M M, Tardós J D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System[J]. IEEE Transactions on Robotics, 2015, 31(5):1147-1163.
[3] Salehi A, Gay-Bellile V, Bourgeois S, et al. Improving constrained bundle adjustment through semantic scene labeling[C]. Springer International Publishing Switzerland 2016,2016: 133-142.
[4] An L F, Zhang X Y, Gao H B, et al. Semantic segmentation-aided visual odometry for urban autonomous driving[J].International Journal of Advanced RoboticSystems,2017,14( 5): 1-11.
[5] Papon J, Schoeler M. Semantic pose using deep networks trained on synthetic RGB-D (2016).
[6] Mur-Artal R, Tard · S J D. ORB-SLAM2: an open-source slam system for monocular, stereo, and RGB-D cameras[J].IEEE Transactions on Robotics,2017,33( 5) : 1255-1262.
[7] Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition[J].Computer Science,2014:1409-1566.
[8] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS (2015).
[9] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
[10] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: CVPR (2016).
[11] Szegedy, C., Reed, S., Erhan, D., Anguelov, D.: Scalable, high-quality object detection. arXiv preprint v3 (2015).
[12] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015).
[13] Liu, W., Rabinovich, A., Berg, A.C.: ParseNet: looking wider to see better. In: ILCR (2016).

[14] Bell, S., Zitnick, C.L., Bala, K., Girshick, R.: Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks. In: CVPR (2016).

[15] Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: Scalable object detection using deep neural networks. In: CVPR (2014).

[16] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: integrated recognition, localization and detection using convolutional networks. In: ICLR (2014).