An Improved ORB Feature Extraction Algorithm

Chuanping Geng¹, Junjie Yang¹,², Jinchen Lin¹, Ting Yu¹, Kexin Shi¹

¹Shanghai Dianji University, Electronic Information College, Shanghai 200000
²Shanghai University of Electric Power, College of Electronic and Information Engineering
1373935106@qq.com

Abstract. In this paper, an improved ORB feature extraction algorithm is proposed to solve the insufficient detection effect of the traditional ORB algorithm on the image edges as well as the low actual extraction of feature points. First, the image pyramid layering was established, and then on basis of the improved oFAST12-20 corner point algorithm, the method with a setting dynamic threshold, according to the idea of adaptive threshold, was proposed to extract feature points, and the candidate set of feature points was obtained. Then, Harris algorithm was used to calculate the response value of each candidate point, and the response value was sorted from high to low. Finally, the top 80% of the feature points with high response value were extracted as the most important feature points. The experimental results show that the improved oFAST algorithm improved the extraction rate of feature points and reduces the algorithm time compared with the traditional ORB corner detection.

1. Introduction

Intelligent robots have gradually appeared in various industries with the advent of the intelligent era. Therefore, simultaneous localization and mapping (SLAM) have great applications and research value for intelligent robots. More and more research on positioning and composition is gradually focused on visual SLAM. The key research problem of visual SLAM is how to find representative image features from images [1]. Image feature extraction and matching is an extremely important step for the entire visual SLAM research, and it is of great significance to target recognition and image reconstruction. From the current research, the common feature point extraction algorithms are Scale Invariant Feature Transform (SIFT) [2], Speeded-Up Robust Features (SURF) [3] and Oriented FAST and Rotated BRIEF(ORB) [4].

Compared with SIFT and SURF, the ORB feature has a calculation speed of 100 times SIFT and 10 times SURF [5]. The literature [6] proposed an ORB algorithm based on modular region segmentation. Although the algorithm ensures that the feature points can be extracted uniformly, it cannot change the number of extracted feature points, and the extraction speed is slower than the traditional ORB algorithm. The literature [7] proposed an ORB algorithm for region division. The ORB algorithm solved the problem of overlapping and redundancy of feature points, but it was not ideal for edge detection.

2. ORB algorithm principle

The ORB feature is an algorithm for quickly extracting and describing feature points. It consists of two parts: feature point extraction and feature point description.
### 2.1 Feature point extraction

In the existing ORB algorithm, FAST [8] corner point algorithm is used to extract feature points. The disadvantage of the FAST corner algorithm is that it does not have directionality and scale invariance, so the FAST detector is improved and optimized in the ORB algorithm to increase its rotation invariance, which is called (Oriented FAST, oFAST). The central idea of FAST feature point extraction is that: if a pixel differs greatly from its surrounding pixels (too bright or too dark), then it may be a feature point.

The steps of oFAST feature point extraction are as follows:

**Step 1.** Build an image pyramid to ensure that the feature points have scale invariance;

**Step 2.** Use the FAST algorithm to preliminarily extract the corner points of each layer of the pyramid image. First, select a pixel \( P \) to determine whether it is a feature point. Select 16 pixels on a circle with a fixed radius of 3 at the center of \( P \), and take the absolute value \( N \) as the difference from the center pixel \( I(p) \), as in formula (1). Choose a suitable threshold value \( t \) (such as 20% of \( I(p) \)), when there are consecutive \( n \) points (\( n \) can be 9 or 12) among these 16 pixels and the gray value of pixel \( P \) If the absolute value of the difference \( N \) is greater than or less than \( t \), it is considered as a characteristic point. A pre-judgment process is added to the FAST-12 algorithm to accelerate the feature point screening rate. Detect the 1,5,9,13th pixel value on the neighborhood circle. Only when the absolute value of the difference between three of the four pixel values and the pixel value of \( P \) point is greater than or less than \( t \), the point \( P \) may become a feature point.

\[
N = |I(x) - I(p)|, \quad x \in \text{circle}(p) \quad (1)
\]

**Step 3.** Using the determined feature point \( P \) as the origin to establish a plane coordinate system. Within the circle whose center is \( p \) and radius is \( r \), it is defined as neighborhood \( W \), calculate the position of the center of mass within \( W \), use \( P \) as the starting point, and the center of mass as Endpoint construction vector. The moment of the domain \( W \) is:

\[
m_{pq} = \sum_{x,y} x_p y_q I(x,y) \quad (2)
\]

In the formula, \( I(x,y) \) is the gray value at a certain point of the image in the neighborhood. From formula (2), \( x, y \in (-r, r) \) the 0th and 1st moments of the neighborhood \( W \) can be obtained as:

\[
m_{00} = \sum_{x,y} I(x,y) \quad (3)
\]
\[
m_{10} = \sum_{x,y} x I(x,y) \quad (4)
\]
\[
m_{01} = \sum_{x,y} y I(x,y) \quad (5)
\]

From equations 3, 4, and 5, the centroid of the neighborhood \( W \) can be derived as:

\[
\mathbf{C} = \begin{pmatrix} m_{10} \\ m_{01} \\ m_{00} \end{pmatrix} \quad (6)
\]

The direction angle of oFAST feature point is

\[
\theta = \tan^{-1}(m_{01}, m_{10}) \quad (7)
\]

Ensure that it improves the rotation invariance of feature points, and ensure that \( x, y \) are within a circle with radius [9].

**Step 4.** Screen the extracted feature points and calculate the response value of the corner points, that is, the sum of the absolute deviations of the gray values of the surrounding pixels and the center point. The obtained response values are sorted from large to small, excluding the point where the response value is small[10]. The algorithm steps are as follows:

1) Take one corner of the image as the origin \( O \), establish the coordinate system \( O-XY \), and calculate the gradient \( I_x, I_y \) of \( I(x, y) \) in the X and Y directions;

2) Calculate the product of two gradient directions \( I_{x2}, I_{y2} = I_x * I_y \);

3) Use Gaussian function to perform Gaussian weighting \((\sigma=1)\) on \( I_{x2}, I_{y2} \) and \( I_{xy} \) to generate elements \( A, B \) and C of matrix \( M \), respectively:

\[
A = g(I_{x2}^2) = I_{x2}^2 \otimes w \quad (8)
\]
\[
B = g(I_{xy}) = I_{xy} \otimes w \quad (9)
\]
\[
C = g(I_{y2}^2) = I_{y2}^2 \otimes w \quad (10)
\]

4) Calculate the response value \( R = \det M - \alpha(\text{trace} M)^2 \), among them \( \det M = AC - B^2 \).
TraceM=A+C.

2.2. Feature point descriptor
The ORB feature uses the BRIEF [11] descriptor, which is a binary coded descriptor. Compared with the method of describing feature points in the area gray histogram, it greatly speeds up the establishment of feature descriptors, and it also greatly reduces the time of feature matching.

The core idea is to select a neighborhood of $S \times S$ around the feature point, take any pair of points $X$, $Y$ in the neighborhood, compare the pixel values of the two points, and if the pixel value of $X$ point is less than $Y$ point, assign 1, Otherwise 0. According to this method, $N$ pairs of points (generally $N$ is 256) are selected in the neighborhood for pixel value comparison to obtaining a string of binary codes. This code is the BRIEF feature descriptor.

3. Improved ORB feature extraction algorithm
The oFAST corner detection algorithm used by the ORB feature is more concentrated on the distribution of the extracted feature points. There are more redundant points and the edge position detection sensitivity is not strong. It ignores the feature point detection of the image edge position. In general, the more feature points are selected, the better the image matching result is, but the excessively dense feature points are very inconvenient for feature description and will affect the matching accuracy [12]. In order to solve the problem of insufficient edge sensitivity and dense feature points, this paper improves the oFAST algorithm. As shown in Figure 1, 20 pixels are selected on the circumference with a $P$ point as the center radius of 3 and the $P$ pixel is set to $I_p$, meanwhile set the adaptive threshold.

![Fig 1 Schematic diagram of improved oFAST corner detection](image)

The algorithm flow is as follows:
Specific steps of the algorithm:

Construct an image pyramid, obtain a set of down-sampled images through Gaussian smoothing and sub-sampling, and scale the image into 8 layers, with a scaling factor of 1.2 for each layer, resulting in a total of 8 images. Set the number of extracted feature points, and then extract the feature points through all the pyramids. Use the improved oFAST feature point extraction algorithm. Harris corner response function is used to remove bad feature points. Finally, output the feature points.
points of each layer of images separately and record the layer where the feature points are located, and accumulate the feature points of a frame of images.

Set the dynamic threshold $T_a$:

$$T_a = \theta \frac{\sum_{i=1}^{20} I_i - I_a}{I_a} \quad (11)$$

Where $\theta$ is the scale factor, the value is taken $\theta = \frac{I_{\text{min}}}{I_{\text{max}}}$. $I_a$ is the gray value at pixel $a$, and $T_a$ is pixel $a$.

Perform feature point detection for each layer of the pyramid. If the pixel value at the detection point $P$ is $I_p$, select 20 pixels in the neighborhood of radius 3, and if there are 12 consecutive points in the 20 pixels satisfy the formula (9), $P$ is considered as a possible feature point.

$$\left\{ \begin{array}{l} |I_a - I_p| > T_a \\ |I_a - I_p| < T_a \end{array} \right. \quad (12)$$

After the detection in step 3, there are some poor quality points in the extracted feature points, which may affect the subsequent matching, so screening is required. In this paper, the Harris algorithm is used to eliminate the poor quality feature points, and the Harris response values of the extracted feature points are calculated respectively, and then the response values are sorted from large to small, and finally, the first $n$ high response value feature points are taken as the final feature point set.

4. Experimental results and analysis
This experiment is implemented on the Ubuntu 14.04 system and Opencv 3.3.1 platform. The selected image is a frame of the image captured in the video taken by the binocular camera. The original image is shown in Fig 3. The characteristic point extracted by the traditional ORB algorithm and the result graphs of the feature points extracted by the mentioned algorithm is shown in Fig 4 and Fig 5.

![Fig 3 Original image](image1)

![Fig 4 The original ORB algorithm results](image2)

![Fig 5 Improved ORB algorithm results](image3)

It can be seen from the figure that compared with Fig 5, the feature points obtained in Fig 4 are more concentrated, and the effect on edge detection is not good. Fig 5 adds the effect of edge detection and improves the extraction of image edge feature points rate.
In terms of performance, the actual number of feature points extracted by the two algorithms and the algorithm time are compared, as shown in the following table.

Table 1 Comparison of the number of feature points extracted

| Algorithm          | Total feature points | Actual feature points | Extraction rate |
|--------------------|----------------------|-----------------------|-----------------|
| ORB algorithm      | 500                  | 396                   | 0.792           |
| Improved ORB       | 500                  | 478                   | 0.956           |

Table 2 Algorithm running time comparison

| Algorithm          | Extract feature points | Total time/ms | Average time/ms |
|--------------------|------------------------|---------------|-----------------|
| ORB algorithm      | 396                    | 33.4          | 0.0843          |
| Improved ORB       | 478                    | 25.6          | 0.0536          |

It can be drawn from Table 1 that when the image pyramid is introduced and the Harris algorithm is used to remove some feature points. The Improved ORB algorithm in this paper improves the feature point extraction rate by 16.4% compared with the traditional ORB algorithm. As can be seen in Table 2, when the number of feature points extracted by the improved algorithm is larger, the time-consuming is faster than the traditional ORB algorithm by 7.8ms. The algorithm proposed in this paper not only improves the extraction rate of feature points but also reduces the arithmetic speed of the algorithm.

5. CONCLUSION

In this paper, it's found that analyzing the effect of traditional ORB algorithm on edge detection is not good and the actual extraction rate is low, an improved ORB feature extraction algorithm is proposed to establish an image pyramid. So ofAST12-20 algorithm is used to set a dynamic threshold to initially extract feature point sets. And then use Harris responding to culling the poor feature points to get the final feature point set. The algorithm in this paper improves the actual extraction rate and reduces the algorithm's time. The experimental results show that the algorithm in this paper is superior to the traditional ORB algorithm in feature point extraction, the operation speed is improved, and it has better robustness.

Acknowledgments

This paper was supported by Shanghai Technology Innovation Project (17020500900), and “Shuguang Program” sponsored by Shanghai Education Development Foundation and Shanghai Municipal Education Commission(17SG51).

References

[1] Gao Xiang, Zhang Tao. Vision SLAM Chapter 14: From Theory to Practice [M]. Beijing: Electronic Industry Press, 2017: 10-14(in Chinese).

[2] David G. Distinctive image features from scale invariant keypoints [J]. International Journal of Computer Vision, 2004, 60 (2) : 91-110.

[3] Herbert B., Andreas E., Tinne T., et al. Speeded-up robust features[J]. Computer Vision and Image Understanding, 2016, 110 (3) : 404-417.
[4] RUBLEEE, RABAUDV, KONOLIGEK, et al. ORB: An efficient alternative to SIFT or SURF [C] // IEEE International Conference on Computer Vision, Barcelona, Spain, 2011: 2564 – 2571.

[5] LIU W, QIAN L. Comparative analysis of SIFT and SURF and ORB algorithms based on OpenCV environment[J]. Control and Instruments in Chemical Industry, 2018, 45(9): 714-716 (in Chinese).

[6] PAN P. An improved ORB algorithm[J]. Microcomputer & Its Applications, 2017, 36(20): 23-26 (in Chinese).

[7] Sun Hao, Wang Peng. An improved ORB algorithm based on region division. Journal of Beijing University of Aeronautics and Astronautics (in Chinese).

[8] VISWANATHAN D G. Features from accelerated segment test (FAST) [EB/OL] . (2016-04-15) [2017-03-30]. https://wenku.baidu.com/view/af9743d725c52cc58ad6beac.html.

[9] Yang Bingkun, Cheng Shuying, Zheng Qianying. Improved ORB feature matching algorithm[J]. Sensors and Microsystems, 2020, 39(02): 136-139 (in Chinese).

[10] yangleo1987. Feature point detection learning _3 (Harris algorithm) https://blog.csdn.net/yangleo1987/article/details/53261637. 2016-11-21/2020-06-20

[11] CALONDER M, LEPETIT V, STRECHA C, et al. Brief: binary robust independent elementary features[C]. European Conference on Computer Vision, Springer Berlin Heidelberg, 2010: 778-792.

[12] Li Yufeng, Li Guangze, Gu Shaohu, Long Kehui. Image stitching algorithm based on regional block and scale-invariant feature transformation [J]. Optical Precision Engineering, 2016, 24(05): 1197-1205 (in Chinese).