WEP Warwick Evans Publishing

International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 2, Number 2, Year 2024
DOI: https://doi.org/10.62051/ijcsit.v2n2.15
Journal homepage: https://wepub.org/index.php/IJCSIT/index

# Implementation of A Fast ORB Keypoints Extraction Algorithm Based on Grid Division

Yuan Cao[1, *], Jun Lu[2], Xu Zhang[3]

[1]School of Software Engineering, Chengdu University of Information Technology, Chengdu, 610000, China

[2]School of Software Engineering, Chengdu University of Information Technology, Chengdu, 610000, China

[3]School of Software Engineering, Chengdu University of Information Technology, Chengdu, 610000, China

*Corresponding Author: 1163020824@qq.com

## ABSTRACT

Synchronous Localization and Mapping (SLAM) is a research hotspot in the field of mobile robots. However, visual SLAM faces a high computational burden in feature point extraction and other processes, thus requiring powerful computing and memory resources as support. This article proposes a fast ORB feature point extraction method based on grid partitioning to meet the requirements of fast and efficient feature point extraction in visual SLAM. This method can significantly reduce the time required for feature point extraction and also meet the requirement of uniform distribution of feature points in images in visual SLAM. Finally, in the experiment, the algorithm proposed in this paper was compared with the traditional ORB feature point extraction algorithm, verifying that this method has significant advantages in extraction speed.

## KEYWORDS

Keypoints extraction; Visual SLAM; ORB features

## 1. INTRODUCTION

Keypoints extraction is one of the important steps in visual SLAM [1]. In visual SLAM, it is usually necessary to extract hundreds of keypoints in one frame of the image to ensure that there are enough points to successfully match the points in the previous frame. And these keypoints should meet the characteristics of low redundancy and uniform distribution, because the map built by clustered and redundant keypoints will have road marking points clustered in some features, which is not conducive to pose calculation and can easily lead to tracking loss. Therefore, the research difficulty of this project is how to achieve fast and uniform keypoints extraction of images under low cost and low computing power hardware conditions.

The process of keypoints extraction is to find a region in the image that has significant texture or brightness differences. Classic image feature extraction algorithms include Scale Invariant Feature Transform (SIFT) [2], Accelerated Robust Feature Transform (SURF) [3], Fast Feature Point Extraction and Description (ORB) [4], and so on. The ORB feature extraction algorithm has the advantage of fast extraction speed. Experiments have shown that the ORB feature extraction speed is about 50 times that of SIFT and 10 times that of SURF [5]. This is due to the use of FAST algorithm in ORB algorithm to extract corner points. Compared to other corner detection algorithms, FAST

corner points only compare pixel brightness, which makes its detection speed very fast. Therefore, using the ORB algorithm for feature extraction on the robot end is undoubtedly the most suitable.

In addition to the speed of keypoints extraction, the uniformity of its distribution in the image is also an important indicator of feature extraction quality. Directly extracting keypoints from the image without using other optimization methods can result in excessively dense keypoints and a large amount of redundancy, thereby affecting the pose estimation accuracy of SLAM. Therefore, the Adaptive Threshold Quad Tree method [6] is used in the ORB-SLAM [7] system to improve the problem of keypoints density. However, using the quadtree homogenization algorithm to extract ORB keypoints greatly increases the time cost. On the one hand, traversing the entire image consumes a lot of computational time, and on the other hand, recursive quadtree partitioning adds additional computational burden.

Based on the computational burden caused by the above factors in the feature extraction process, this paper proposes a fast ORB feature extraction method based on grid partitioning, achieving the following two objectives:

(1) Using non traversal methods to extract keypoints significantly reduces the computational time of feature extraction;

(2) By dividing the original image into grids and extracting features from the grid, it ensures that the extracted keypoints can be evenly distributed in the image.

## 2. ALGORITHM PROCESS

The fast and uniform feature extraction algorithm based on grid partitioning is mainly divided into two stages, namely the image grid partitioning stage and the ORB feature extraction stage. The purpose of the image grid division stage is to divide the original image into several sub regions based on the number of keypoints to be extracted, providing a detection range for corner points in the subsequent feature extraction process. In the feature extraction stage, a non traversal pixel scanning method is used in each divided grid to extract a corner, and a corner compensation algorithm is used to compensate a keypoint in adjacent grids for grids without corner points. Subsequently, the direction and descriptors of all keypoints are calculated. The process of this algorithm is shown in the following figure 1.

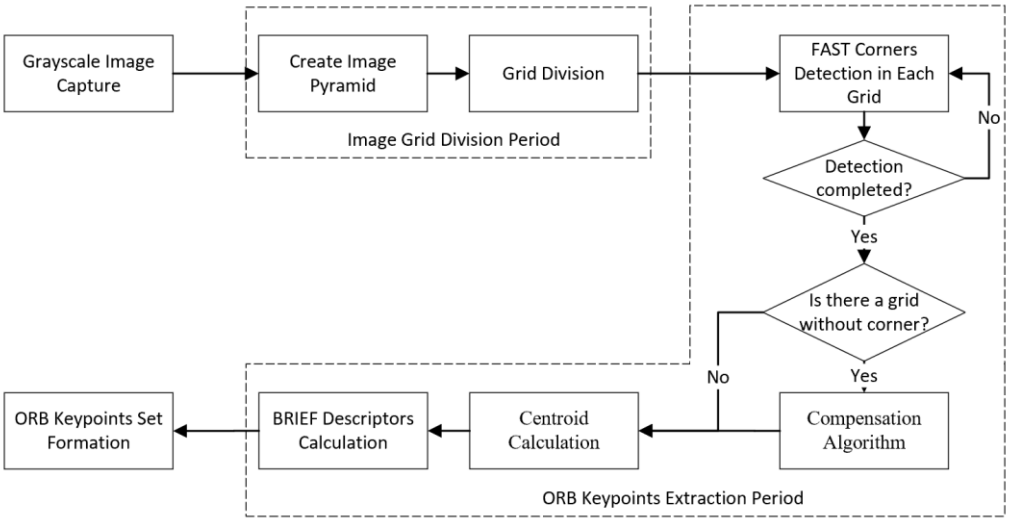


**Figure 1.** The process of fast ORB keypoints extraction algorithm based on grid division

## 2.1. Grid Division

The main task of grid partitioning is to divide the original image into its corresponding number of blocks based on the number of keypoints that need to be extracted. In order to ensure that the extracted key points have scale invariance, the first step is to preprocess the image, and the key to preprocessing is to construct an image pyramid. In this way, the lower level images in the image pyramid can be seen as scenes observed from a distance, while the higher level images are scenes observed from a close range. The structure of image pyramid is shown in figure 2.

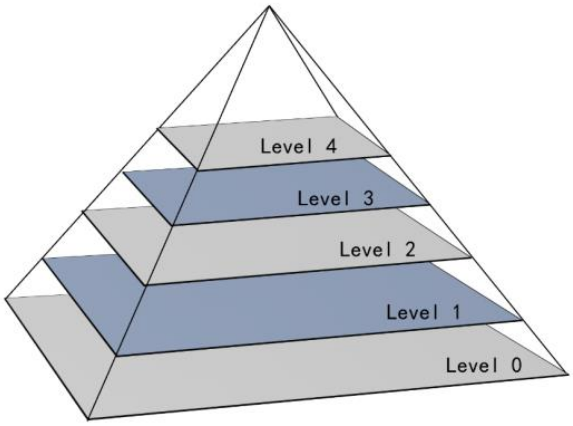


**Figure 2.** The structure of image pyramid

If the number of keypoints to be extracted is T, the image pyramid is divided into grids based on the number of keypoints extracted, resulting in blocks of the same size. The specific process is shown in figure 3.

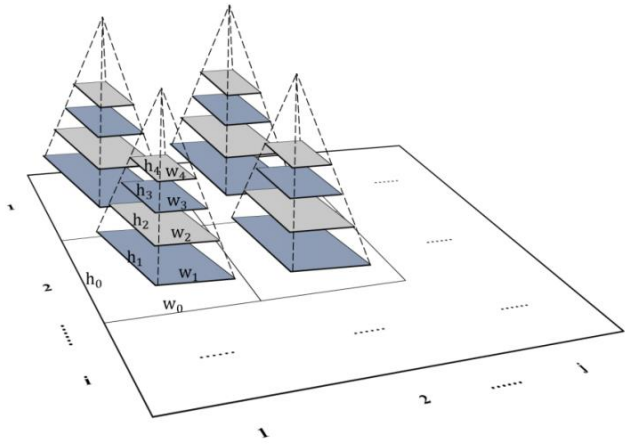


**Figure 3.** Perform grid division on image pyramids

If the resolution of the original image is, the value of and can be determined by the following formula:

$$\begin{cases} i \times j = T \\ \frac{i}{j} = \frac{H}{W} \end{cases} \tag{1}$$

## 2.2. Single FAST corner extraction

After performing grid division on the image pyramid, it is necessary to find a FAST corner point in each sub image pyramid, and after finding the first corner point, no subsequent pixels in the image

pyramid will be detected. Instead, the corner point will be directly detected in the next sub image pyramid.

The extraction of ORB keypoints requires the use of FAST corner detection algorithm to determine the position of keypoints. The detection process is as follows:

① Select a pixel from the image and calculate its grayscale as;

② Manually setting an initial grayscale difference threshold;

③ Take 16 pixel points on the circumference with a point as the center and 3 pixels as the radius. The grayscale values of these 16 pixel points are set to, with values ranging from 1 to 16;

④ Divide these 16 pixels into three categories according to the following formula. The class represents that the grayscale value of the current pixel is much larger than that of the central pixel, which belongs to the bright spot; Class represents that the grayscale interpolation between the current pixel and the center pixel is not significant; Class represents that the current pixel is much darker than the center pixel, and belongs to the category of dark spots;

$$Classify \begin{cases} b, & I_i - I_p \geq t \\ s, & -t < I_i - I_p < t \\ d, & I_i - I_p \leq t \end{cases} \tag{2}$$

⑤ If there are continuous n=12 class points or class points among these 16 pixels, that is, if there are continuous three-quarters of the pixel grayscale and the absolute value of the difference on the circumference of is greater than the threshold, it will be judged as a feature point.

## 2.3. Compensation for the number of points

In grids where the grayscale changes in the image are not significant and the texture is not rich enough, it may not be possible to extract any FAST corners, resulting in the final number of corners being less than the number of corners that need to be extracted. In this case, additional corners need to be extracted in other grids. The core idea of corner compensation is to take the grid that has not been extracted as the center, and try to extract additional corner points from adjacent grids within a distance of, and scan the grids within the neighborhood in an order from inside out.

The final extracted keypoints are shown in figure 4. The black dots in the figure represent the corners in the first grid division, and the red asterisks represent the corners compensated in their neighborhood through the compensation algorithm.

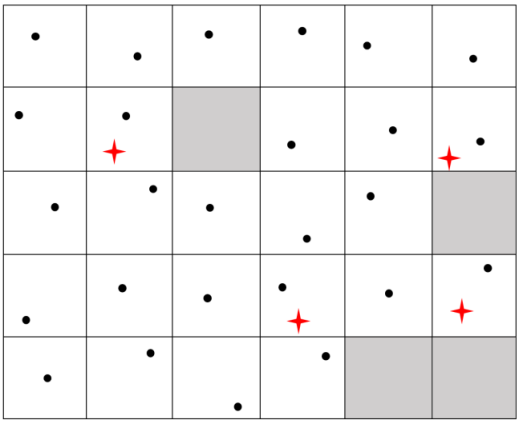

**Figure 4.** The result of compensating for the number of FAST corners

### 2.4. ORB feature calculation

The ORB feature is composed of the coordinates, directions, and feature descriptors of the FAST corners. So the next task is to calculate the direction and feature descriptors of these FAST corners to form a complete ORB feature.

The ORB algorithm uses the grayscale centroid method to calculate the direction of keypoints. If P is the coordinate of the keypoint and Q is the centroid of the grayscale to be calculated, then the line connecting P-Q is the direction of the keypoint. The calculation method of grayscale centroid Q is shown in the following equation:

$$Q = (C_x,\ C_y) = \left(\frac{\sum_{x=-R}^{R}\sum_{y=-R}^{R} xI(x,y)}{\sum_{x=-R}^{R}\sum_{y=-R}^{R} I(x,y)},\ \frac{\sum_{x=-R}^{R}\sum_{y=-R}^{R} yI(x,y)}{\sum_{x=-R}^{R}\sum_{y=-R}^{R} I(x,y)}\right) \tag{3}$$

Then the angle of vector $\overrightarrow{PQ}$ is the direction of the corner, and its calculation formula is as follows.

$$\theta = \arctan\left(C_x/C_y\right) \tag{4}$$

The ORB feature uses the BRIEF(Binary Robust Independent Element Features) algorithm [8] to calculate the description information of keypoints, which has the characteristics of fast speed and low memory consumption. The specific calculation steps for the BRIEF descriptor are as follows:

① Take a neighborhood with a size of S×S centered on the keypoints;

② Select n pairs of points in the neighborhood and define $\tau$ as follows. Among them, I(x) and I(y) are the grayscale values of two pixels in a point pair.

$$\tau(x,y) = \begin{cases} 1 & if\ I(x) < I(y) \\ 0 & otherwise \end{cases} \tag{5}$$

③ Finally, the binary code strings obtained in step 2 are combined to form an N-dimensional vector, which is the feature descriptor of the point. The definition is as follows:

$$f(p) = \sum_{1\le i\le 256} 2^{i-1}\tau(x_i,\ y_i) \tag{6}$$

## 3. EXPERIMENT AND ANALYSIS

To start with, the experiment compared the performance of the fast ORB feature extraction algorithm based on grid division (referred to as the grid method) proposed in this project with two commonly used ORB feature extraction algorithms, namely the unoptimized ORB algorithm (referred to as the ORB method) and the quadtree homogenization extraction algorithm (referred to as the quadtree method) used in ORB-SLAM. Figure 5 is the comparison of keypoints extraction in laboratory environment.

(a) ORB method (b) Quad-tree method (c) Grid method

**Figure 5.** Comparison of keypoints extraction in laboratory environment

Then, figure 6 is the comparison of keypoints extraction in aisle environment.

(a) ORB method (b) Quad-tree method (c) Grid method

**Figure 6.** Comparison of keypoints extraction in aisle environment

At last, figure 7 is the Comparison of keypoints extraction in hallway environment.

(a) ORB method (b) Quad-tree method (c) Grid method

**Figure 7.** Comparison of keypoints extraction in hallway environment

From the comparative experiment above, it can be seen that the distribution uniformity of keypoints is the worst in ORB method. The keypoints are too concentrated, which makes it difficult to accurately describe the spatial information of the image. Both the grid method and the quadtree method are significantly better than the ORB method, but in areas with rich image textures, the keypoints of the quadtree method are also prone to clustering. The fast grid method can disperse the keypoints as much as possible, and is slightly stronger in uniformity than the quadtree method.

Finally, there is a comparative experiment on keypoints extraction speed. The processor used in this experiment is I5-7200U and the memory is 8GB. The comparison results of extraction speed are shown in table 1. From the comparison results, it can be seen that the ORB method and the quadtree method take longer. Although the quadtree method can ensure a certain degree of uniformity in the distribution of keypoints, the process of uniformly selecting keypoints sacrifices a certain amount of time, making the total time slightly higher than the ORB method. Thanks to the non traversal keypoint extraction method adopted by the fast grid method, pixel scanning of the entire image is avoided, resulting in significantly lower average time compared to the other two methods.

**Table 1.** Comparison of average time consumption of keypoints extraction algorithms

| Keypoints Count | ORB Method | Quad-tree Method | Grid Method |
|---|---|---|---|
| 100 | 9.9ms | 14.7ms | 2.7ms |
| 200 | 10.1ms | 14.9ms | 3.8ms |
| 300 | 10.3ms | 14.9ms | 4.3ms |
| 400 | 10.5ms | 15.1ms | 4.9ms |

## 4. SUMMARY

This article first analyzes the existing feature point extraction algorithms and concludes that these algorithms require a longer processing time and are not suitable for use on hardware platforms with lower computational power. Therefore, a fast ORB feature point extraction algorithm based on grid partitioning is proposed, which can achieve both uniform distribution of feature points to meet the requirements of visual SLAM and fast execution,with high application value.

## REFERENCES


[1] Takashi Tsubouchi. Introduction to Simultaneous Localization and Mapping. Journal of Robotics and Mechatronics, 2019, 31(3):367-374.

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

[3] Bay H, Tuytelaars T, Van Gool L. “Surf: Speeded up robust features”, Computer Vision–ECCV 2006, pp. 404–417, Springer.

[4] Rublee E, Rabaud V, Konolige K, Bradski G. “Orb: an efficient alternative to sift or surf”. 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2564–2571,

[5] Suo Chunbao, Yang Dongqing, Liu Yunpeng. “Compare SIFT, SURF, BRISK, ORB, and FREAK algorithms from multiple perspectives”. Beijing Surveying and Mapping 2014 (04): 22-26

[6] MUR-ARTAL R, TARDÓS J D. “ORB-SLAM2: an opensource SLAM system for monocular, stereo, and RGB-D cameras. IEEE transactions on robotics, 2017, 33(5): 1255–1262

[7] Li Guojun, Xu Yanhai, Duan Jiewen, Han Shilei. Extracting ORB-SLAM keypoints using local adaptive threshold method. Bulletin of Surveying and Mapping, 2021(09):32-36

[8] Calonder M, Lepetit V, Strecha C, Fua P. Brief: Binary robust independent elementary features. European conference on computer vision, pp. 778–792, Springer, 2010

[9] Mur-Artal R, Montiel J, Tardós J D. Orb-slam: a versatile and accurate monocular slam system. IEEE T ROBOT, 2015. 31(5):1147-1163.