A REVIEW OF FEATURE DETECTION AND MATCH ALGORITHMS FOR LOCALIZATION AND MAPPING

Shimiao Li *

**School of Precision Instrument and Opto-Electronics Engineering, Tianjin University, Tianjin 300110, China*

Keywords: localization, mapping, vision, feature detection, robot

Abstract

Localization and mapping is an essential ability of a robot to keep track of its own location in an unknown environment. Among existing methods for this purpose, vision-based methods are more effective solutions for being accurate, inexpensive and versatile. Vision-based methods can generally be categorized as feature-based approaches and appearance-based approaches. The feature-based approaches prove higher performance in textured scenarios. However, their performance depend highly on the applied feature-detection algorithms. In this paper, we surveyed algorithms for feature detection, which is an essential step in achieving vision-based localization and mapping. In this pater, we present mathematical models of the algorithms one after another. To compare the performances of the algorithms, we conducted a series of experiments on their accuracy, speed, scale invariance and rotation invariance. The results of the experiments showed that ORB is the fastest algorithm in detecting and matching features, the speed of which is more than 10 times that of SURF and approximately 40 times that of SIFT. And SIFT, although with no advantage in terms of speed, shows the most correct matching pairs and proves its accuracy.

1 Introduction

Localization and mapping is an essential ability of a robot to keep track of its own location in an unknown environment. For any robot or mobile device, avoiding dangerous situations such as collisions and unsafe conditions comes first. What's more, if the robot has a purpose that relates to specific places in the environment, it must find those places. As a result, the technology of mobile robot navigation has become a central topic of robot researches, which is also known as ego-motion estimation or Simultaneous Localization and Mapping (SLAM).

Existing methods for localization and mapping are various: global position system (GPS), inertial navigation system (INS), laser/ultrasonic sensor method and the vision-based technique. However, each of them has its weakness and limitations. GPS can only be used in outdoor circumstances and provides positions with error. The INS is a relative positioning technique that calculates the position by performing mathematical integration with respect to time, leading to drift accumulation. The laser sensor method shows better accuracy but is very expensive. And the efficiency of the ultrasonic method depends largely on the material and surface. Compared with the methods discussed above, vision-based method is a more effective solution for localization and mapping of robots for being more accurate, inexpensive and versatile. It is less prone to the interferences that other sensor-based localization systems encounter and becomes popular in robot navigation.

Vision-based methods can generally be categorized as feature-based approaches, appearance-based approaches and a combination of the two. Particularly in textured scenarios, the feature-based approaches prove higher performance, which depends highly on the applied feature-detection algorithms. However, as feature-based approaches work by analyzing recognizable features inside or between the acquired images. Their overall performance depends largely on the success of feature-detection.

In this paper, we give a review of some feature-detection methods. We will introduce the mathematic models of these methods and compare the performances on feature detection and feature match. Readers who want to get access to object robot navigation, motion estimation or object tracking can learn application field of each feature-detection method from this review.

The rest of the paper is organized as follows. In section Ⅱ, we present the principles of different feature-detection algorithms and focus mainly on how they describe features. To show the performances and application fields of the mentioned algorithms, we conduct a series of experiments and make some discussions in section Ⅲ. Finally, we make some conclusions in section Ⅳ.

2 Feature-detection and match algorithms

Features of an image typically refers to the interesting part with meaningful information for computer vision tasks. They can generally be categorized as low level features and high level ones which come into being with the emergence of more computational problems and time constrains. Feature detection and match are tow necessary steps in most computer

vision tasks. To achieve this, various algorithms have been developed by researchers.

2.1 Corner Detection Algorithms

Corner is a commonly used classic low-level feature characterized by the intersection of two edges that represents a variation in the gradient in the image. Over changes of viewpoint, corner detection generally obtains more stable features over changes of viewpoint than patch matching method [2]. Two representative algorithms in corner detection are the Harris Detector and the Features from Accelerated Segment Test (FAST) method.

2.1.1 Harris Corner Detector

In 1988, Harris and Stephens proposed a classic corner detector which extracts features by looking for points with stability, repeatability and low self-similarity [3]. Suppose a grayscale image I . The variation of intensity with a window $w(x,y)$ and a displacement (u,v) can be calculated and further expressed in a matrix form after Taylor Expansion and proper cancelling, as is shown in (1):

$$\begin{aligned} E(u,v) &= \sum_{x,y} w(x,y)[I(x+u,y+v)]^2 \\ &\approx [u,v](\sum_{x,y} w(x,y)\begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix})\begin{bmatrix} u \\ v \end{bmatrix} \\ &\approx [u,v]M\begin{bmatrix} u \\ v \end{bmatrix} \end{aligned} \quad (1)$$

The response of the detector is

$$R = \det(M) - k(trace(M))^2 \quad (2)$$

Where $\det(M) = \lambda_1\lambda_2$ and $trace(M) = \lambda_1 + \lambda_2$. A window with a score R greater than a certain value is considered a "corner".

2.1.2 The Features from accelerated segment test (FAST) corner detector

The FAST algorithm is originally proposed by Rosten and Drummond for identifying interest points [4]. Considering a pixel P with intensity Ip in the image, n contiguous pixels out of 16 need to be above or below Ip by the selected threshold T (Its authors used $n = 12$ in the first version of FAST) if the pixel is an interest point. To make it faster, the pixels 1, 5,9,13 are compared with Ip first and the other pixels are check only if at least 3 of them are above or below $Ip + T$. The comparatively insignificant points can be discarded by applying non maximal suppression.

The FAST algorithm is developed for real time applications with limited computational resources, like the mobile SLAM.

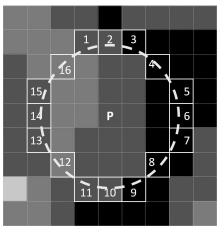

Figure 1: Point P and its 16 surrounding pixels

2.2 Scale-Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF)

Scale-Invariant Feature Transform (SIFT)[5], proposed by Lowe (1999) as a new method of image feature generation, is a vision-based algorithm to detect and describe the scale-invariant and rotation-invariant region-based features in the image. The SIFT features of an image are generated by identifying repeatable points in a pyramid of scaled images and by detecting maxima and minima in the difference-of-Gaussian(DOG) pyramid[6], as is shown in Figure 2.

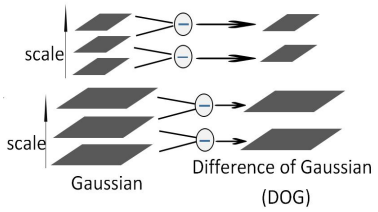

Figure 2: Gaussian pyramid and DOG pyramid

$$\begin{aligned} D(x,y,\sigma) &= (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) \\ &= L(x,y,k\sigma) - L(x,y,\sigma) \end{aligned} \quad (3)$$

Where $I(x,y)$ is the original image, $G(x,y,\sigma)$ is the Gaussian filter at scale σ , $L(x,y,\sigma)$ is the convolution of the image I with the Gaussian blur.

After the low-contrast candidate points and edge response points are discarded by fitting a three-dimensional quadratic function, SIFT descriptors are obtained by attaching orientation parameters m and θ to every feature point with position (x,y) .

$$\begin{aligned} m(x,y) &= \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \\ \theta(x,y) &= a\tan 2(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}) \end{aligned} \quad (4)$$

SIFT is widely used in fields like object recognition, robot localization and mapping, panorama and ego-motion estimation for being relatively easy to extract, distinctive and robust to changes in illumination, noise, and minor changes in viewpoint[7][8]. The application to robot localization and mapping proposed by Lowe [9] uses a trinocular system where SIFT features are extracted and matched through a 'the right to left match' and a refinement using the top

image. The matched features serve as natural landmarks in unmodified environments.

The Speeded up Robust Features (SURF) method [10], first presented by Herbert Bay et al. in 2006, is a robust local feature detector partly inspired by SIFT but works differently in some aspects. SURF features are extracted using Hessian Matrix computed on every pixel followed by a non-maximal suppression [11] process:

$$H(x,\sigma)=\begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \quad (5)$$

Where L is defined similarly and and Lxx, Lxy, Lyy are the second partial derivatives of L . It uses an integer approximation to the determinant of Hessian blob detector and uses the sum of the Haar wavelet response around the interest point instead of the gradient histogram for feature descriptors.

SURF is a feature-detection algorithm used in computer-vision tasks and proves faster and more robust against different image transformations than SIFT [12].

2.3 The Binary Robust Independent Elementary Features (BRIEF) method

The BRIEF algorithm [13] is proposed by Calonder in 2010 as a fast binary descriptor. In the BRIEF algorithm, interest points can be extracted using FAST, HARRIS, SIFT or SURF. After the denoising process using Gaussian filtering, the descriptor is obtained by selecting N pairs ($N=125$) of features points (which are subject to Gauss distribution) from the neighborhood of an interest point and getting binary values after comparison between the two intensities in each feature pair $< P_1, P_2 >$. The N binary values are put together to form an N-dimensional binary encoding.

BRIEF can achieve fast descriptor-building and feature-matching. The algorithm builds descriptors by detecting random response instead of by using traditional gradient histogram and simply calculate the Hamming distance to match features. However, the drawback is that BRIEF has poor rotation invariance and is limited to conditions with small rotation.

2.4 Oriented FAST and Rotated BRIEF algorithm (ORB)

The ORB algorithm [14] is first presented by Ethan Rublee et al. in 2011 as a method based on FAST and BRIEF. Interest points are extracted from images using the FAST algorithm and descriptors are obtained using the BRIEF method.

As an improved version based on FAST and BRIEF, the ORB algorithm achieved better rotation invariance by establishing a 2 dimensional system with the key point as its center and the line connecting the key point and the centroid of the connection area as its x axis,as is shown in Fig. 3:

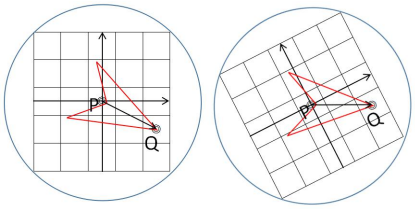

Fig. 3 The 2 dimensional system in ORB to achieve rotation invariance

$$M_{00}=\sum_{X=-R}^{R}\sum_{Y=-R}^{R}I(x,y) \qquad M_{10}=\sum_{X=-R}^{R}\sum_{Y=-R}^{R}xI(x,y)$$
$$M_{01}=\sum_{X=-R}^{R}\sum_{Y=-R}^{R}I(x,y) \qquad Q_X=\frac{M10}{M00}, Q_Y=\frac{M01}{M00} \quad (6)$$

The ORB algorithm is a fast method in feature detection, benefiting from the FAST algorithm, but fails to solve the problem of poor scale invariance. Researches proves that the speed of ORB is more than 100 times that of SIFT and more than 10 times that of SURF. And in actual image and video processing, strategies like the usage of image pyramid are used to improve scale invariance.

2.5 Others

Some other algorithms are also used for feature-detection and each of them has its own characteristics. The Binary Robust Invariant Scalable Keypoints (BRISK) method [15] is an improved version of BRIEF with better robustness against noises, scale invariance and rotation invariance. The Maximally Stable Extremal Regions (MSER) method [16] was proposed in 2002 and proves useful by its author in robust wide baseline stereo. The Good Features to Track (GFTT) [17] method determines strong corners in images. The Fast Retina Keypoint (FREAK) method [18] is based on the distribution of human retinal cells which is dense in the middle and sparse in the surrounding to acquire points. The pixels of each region of points are randomly compared to get a set of binary values as descriptors.

3 Experiments and Results

3.1 Scale invariance

Due to the move of the robot or objects, images in adjacent frames taken by the camera on the device may vary a lot in scale, leading to the invalidity of some feature-detection methods.

In our first experiment, we tested the performance of the algorithms on scale invariance detection. We tried different algorithms to find corresponding points in these two images in different scale. A larger number of matching points then indicated a better scale invariance detection. As is shown in

TABLE 1, SIFT and SURF has the better scale invariance among the algorithms.

Method	*Harris*	*FAST*	*BRISK*	*MSER*	*SIFT*	*SURF*	*ORB*
Correct matches	3	0	5	10	121	62	16

Table 1: Correct matches between images in different scale

3.2 Rotation invariance

Similarly, the scenes and objects can also rotate between different frames due to the relative position changing. This paper conducts experiments by rotating the original images at 20 degree and matching them with the original ones. The scale invariance of different algorithms is evaluated by comparing the matching result. As is shown in TABLE Ⅱ, SIFT and SURF has the best rotation invariance.

Method	*Harris*	*FAST*	*BRISK*	*MSER*	*SIFT*	*SURF*	*ORB*
Correct matches	28	20	13	27	279	81	40

Table 1: Correct matches between rotated images

3.3 Speed

Speed is an important indicator in determining the efficiency of a method. Algorithms with high speed is more advantageous under the condition of similar hardware configuration and matching effect.

In this paper, the speed of different feature detection and match is evaluated by comparing each method on the total time consumption of feature detection, description and match. As is shown in TABLE Ⅲ, ORB is the fastest algorithm in detecting and matching features, the speed of which is more than 10 times that of SURF and approximately 40 times that of SIFT.

Method	*Harris*	*FAST*	*BRISK*	*MSER*	*SIFT*	*SURF*	*ORB*
Total time	1.497	0.858	1.513	1.294	3.915	1.326	0.107

Table 3: Total time consumed in detection and match

3.4 Discussions

Finally, we conducted experiments on the overall performance of the algorithms by changing the viewpoint and scale before matching them with original ones.

From the correct match points in different methods obtained in our experiment, it is proved clearly that SIFT, SURF and ORB have better overall performance, while methods(like Harris, FAST, and BRISK) based on grayscale information extracts low-level features and performs poor scale and rotation invariance. According to our experiments, ORB is the fastest algorithm in detecting and matching features, the speed of which is more than 10 times that of SURF and approximately 40 times that of SIFT. And SIFT, although with no advantage in terms of speed, shows the most correct matching pairs and proves its accuracy.

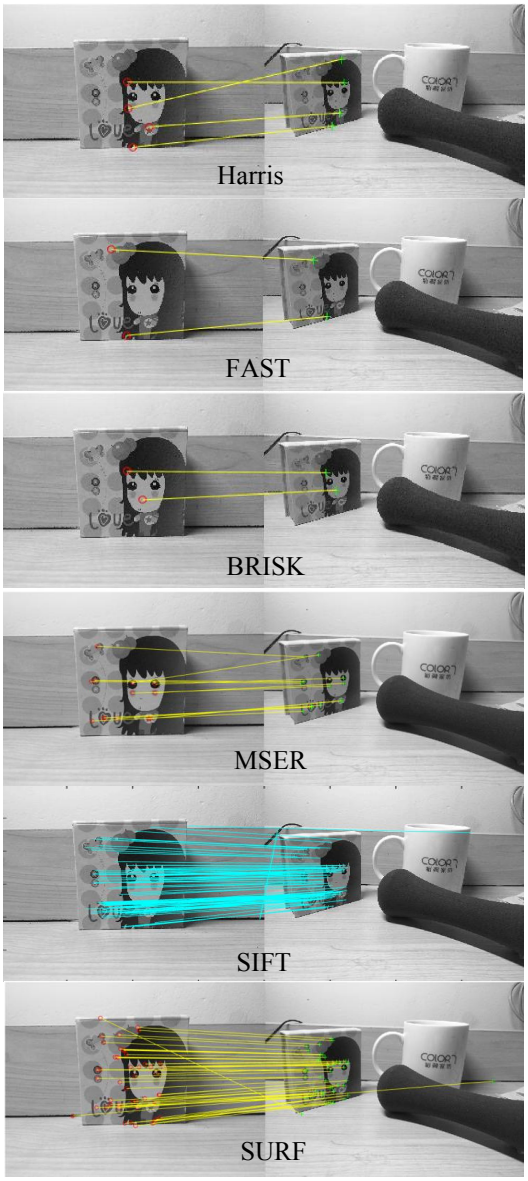

Figure 4: Feature detection and match in different methods

4 CONCLUSION

As feature detection and match are two necessary steps in most computer vision tasks, we give a review on the algorithms of feature detection and match in this paper. We present the mathematical models and introduce the applications of each algorithm. To show the difference of their performances, we conduct a series of experiments and make a discussion about the results.

Some conclusions can be acquired through our review and experiments. Harris and FAST are grayscale-based algorithms with no scale or rotation invariance and show poor performance in feature match. But FAST shows higher speed in detecting features. MSER performs better than Harris and Fast. SIFT, SURF and ORB shows better overall performance and prove suitable in feature match. In terms of speed, the order is ORB, SURF, SIFT, while in terms of accuracy, the

order is SIFT, SURF, ORB. Thus ORB is especially suitable for real-time tasks. SIFT, though with no advantage in terms of speed, is the best option if high accuracy is required. These conclusions can serve as a reference for researches who want to select proper methods for tasks requiring feature extraction and match.

References

[1] Aqel, M. O. A., Marhaban, M. H., Saripan, M. I., & Ismail, N. B. (2016). Review of visual odometry: types, approaches, challenges, and applications. SpringerPlus, 5(1), 1897. http://doi.org/10.1186/s40064-016-3573-7

[2] Barnes, C., Shechtman, E., Finkelstein, A., & Goldman, D. B. (2009). Patchmatch: a randomized correspondence algorithm for structural image editing. Acm Transactions on Graphics, 28

[3] Harris, C. (1988). A combined corner and edge detector. Proc Alvey Vision Conf, 1988

[4] Rosten, E., & Drummond, T. (2006). Machine Learning for High-Speed Corner Detection. Computer Vision 鈥?ECCV 2006

[5] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision

[6] Catarious-Dm, J., & Baydush, A. C. J. (2006). Characterization of difference of gaussian filters in the detection of mammographic regions. Medical Physics, 33

[7] Zhou, H., Yuan, Y., & Shi, C. (2009). Object tracking using sift features and mean shift. Computer Vision & Image Understanding, 113

[8] Wensley, J. H., Lamport, L., Goldberg, J., Green, M. W., Levitt, K. N., & Melliarsmith, P. M., et al. (1989). SIFT: Design and analysis of a fault-tolerant computer for aircraft control. Tutorial: hard real-time systems

[9] Se, S., Lowe, D. G., & Little, J. J. (2002). Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. International Journal of Robotics Research, 21

[10] Bay, H., Tuytelaars, T., & Gool, L. V. (2006). Surf: speeded up robust features. Computer Vision & Image Understanding, 110

[11] Neubeck, A., & Van Gool, L. (2006). Efficient Non-Maximum Suppression. International Conference on Pattern Recognition

[12] Luo, J., & Gwun, O. (2009). A comparison of sift, pca-sift and surf. International Journal of Image Processing, 3

[13] Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2011). BRIEF: Binary Robust Independent Elementary Features. Computer Vision - ECCV 2010, European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings

[14] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. International Conference on Computer Vision

[15] Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011). BRISK: Binary Robust invariant scalable keypoints. International Conference on Computer Vision

[16] Matas, J., Chum, O., Urban, M., & Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. Image & Vision Computing, 22

[17] [19]Shi, J. (1994). Good features to track. IEEE Computer Society Conference on Computer Vision & Pattern Recognition, Cvpr

[18] [20]Ortiz, R. (2012). FREAK: Fast Retina Keypoint. Computer Vision and Pattern Recognition