

PAPER • OPEN ACCESS

Research on Image Feature Point Matching Based on ORB and RANSAC Algorithm

To cite this article: Hua Zhang *et al* 2020 *J. Phys.: Conf. Ser.* **1651** 012187

View the [article online](#) for updates and enhancements.

You may also like

- [Image reconstruction in EIT with unreliable electrode data using random sample consensus method](#)
Min Ho Jeon, Anil Kumar Khambampati, Bong Seok Kim et al.
- [Research on Purifying Paired Feature Point Based on RANSAC Algorithm](#)
Xiaofei Sun, Caihong Ma and Jiali Zhou
- [Image Matching Method Based on Laplacian Feature Constrained Coupling Variance Measure](#)
Hongwei Yang, Yongfeng Qi and Gang Du



ECS
The Electrochemical Society
Advancing solid state & electrochemical science & technology

DISCOVER
how sustainability
intersects with
electrochemistry & solid
state science research

Research on Image Feature Point Matching Based on ORB and RANSAC Algorithm

Hua Zhang^{1,2,3a}, Guoxun Zheng^{1,2,3*} and Haohai Fu^{1,2,3}

¹School of Computer Technology and Engineering, Changchun Institute of Technology, Changchun, Jilin, 130012, China

²Jilin Provincial Key Laboratory of Changbai Historical Culture and VR Reconstruction Technology, Changchun, Jilin, 130012, China

³Jilin Provincial Engineering Research Center of Changbai Ecological Culture Virtual Reality Simulation, Changchun, Jilin, 130012, China

^azhanghua@ccit.edu.cn *Corresponding author's e-mail: citsoft_zgx@qq.com

Abstract. Image matching is one of the basic problems in the field of machine vision research. In order to improve the accuracy of image feature point matching and enhance the anti-interference ability of the algorithm, for the shortcomings of traditional ORB algorithm, an image feature point matching algorithm based on ORB and RANSAC algorithm is proposed. The algorithm can filter out false matches on the basis of coarse matching results, and filter out exact matching points through the homography matrix. Experiments show that this method can effectively improve the accuracy of image matching and shorten the execution time, and it is very robust to image matching with different scales, ambiguities, brightness and rotation.

1.Introduction

Image matching is one of the basic problems in the field of machine vision research, which is widely used in medical images[1], remote sensing images[2-3], face recognition[4], visual SLAM[5] and three-dimensional reconstruction[6] and other fields, through the corresponding relationship among content, feature and structure, the texture and gray level of the image are analyzed, and the similarity and consistency are analyzed to seek similar image targets. In the image matching algorithm, there are many feature extraction and matching methods, SIFT, SURF, BRISK, ORB and so on. Among them, SIFT algorithm was proposed by David Lowe[7] in 1999, and was further developed and improved in 2004. This algorithm is the most robust local feature algorithm, but it has a large amount of calculation and cannot meet the real-time requirements well. To this end, Ehtan Rublee et al. proposed the ORB[8] algorithm in 2011, which uses an improved FAST[9] algorithm and an improved BRIEF[10] algorithm. Both algorithms have the advantage of fast calculation speed, so the ORB is very good at the calculation speed, which is 1 to 2 orders of magnitude faster than the SIFT and SURF algorithms. However, in terms of matching accuracy, the ORB algorithm is relatively weak, and the matching results contain a lot of error information. The matching accuracy is low and the anti-interference ability is poor. In this paper, the ORB and RANSAC algorithms are used to complete the extraction of image feature points and eliminate the false matching. On the basis of ensuring strong real-time processing and fast feature point extraction, the matching accuracy is improved.



2. Image feature point extraction

2.1. Feature point detection

The first step of ORB feature detection is to use the FAST algorithm to find the key points in the image, namely the feature points. The FAST algorithm determines whether it is a key point by comparing the gray levels of pixels around a candidate pixel. With the candidate pixel P as the center and 3 pixels as the radius, a circle is formed, which passes through a total of 16 pixels and is numbered 1 to 16 in the clockwise direction. Given a threshold h , compare the gray value of the P point with the gray value of these 16 pixels. If there are N (generally 12) consecutive pixels and the absolute value of the gray value of the P point is greater than or Equal to h , that is, the brightness is higher or lower than P , then point P is determined as a feature key point. To speed up the determination process, only point P is compared with 4 equally spaced pixel points 1, 5, 9, 13 on the circle. If there are 3 or more pixels whose brightness is higher or lower than P , then P is selected as the key point. This optimization reduces the time to search for key points in the entire image by four times. The judgment formula is as follows:

$$f_{\text{det}}(I_x, I_p) = \begin{cases} 1, & |I_x - I_p| \geq h \\ 0, & |I_x - I_p| < h \end{cases} \quad (1)$$

$$N = \sum_{x \in \text{circle}(P)} f_{\text{det}}(I_x, I_p) \quad (2)$$

Among them, I_p is the gray value of the candidate pixel point P , and I_x is the gray value of the points 1 to 16 around the P point.

In order to solve the direction sensitivity problem of the FAST algorithm, the ORB algorithm uses the improved oFAST algorithm to add direction information to the FAST feature. This direction depends on how the intensity around the key point changes.

The ORB algorithm first finds the intensity centroid in the neighborhood of the feature point, and the location of the average pixel intensity in the given neighborhood, also known as the centroid, and then determines the direction of the key point by the vector direction from the key point to the centroid, and the brightness in the neighborhood Strengthen in this direction. The formula for the neighborhood moment can be expressed as:

$$m_{pq} = \sum_{x,y \in S} x^p y^q I(x,y) \quad (3)$$

Among them, S represents the domain of the key point, and $I(x,y)$ is the gray value at the point (x,y) . The centroid coordinates of the neighborhood can be expressed as:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (4)$$

Among them, m_{00} , m_{01} and m_{10} are the values when p and q in the formula (3) are 0 or 1, respectively. Therefore, the main directions of the feature points are:

$$\theta = \arctan(m_{01}, m_{10}) \quad (5)$$

2.2. Feature descriptor

The second step of the ORB algorithm is to use the BRIEF algorithm to convert the key points found by the FAST algorithm into binary feature vectors, which can collectively represent an object. Binary feature vectors, also known as binary descriptors or descriptors, are feature vectors that contain only 1s and 0s. Each key point in the BRIEF is described by a binary feature vector, which is generally a string of 128-512 bits. The brief algorithm process is as follows:

(1) Use Gaussian filter kernel to smooth the target image (Gaussian filter kernel is set to 9×9 , variance is 2) to prevent the descriptor from being too sensitive to high-frequency noise and increase stability.

(2) Pick a key point.

(3) A pair of pixels are randomly selected from the well-defined neighborhood around the key points selected in the above steps. The neighborhood around the key point is a square with a specific pixel width and height. The first pixel in a random pair is a pixel drawn from a Gaussian distribution centered on key points, with a standard deviation or dispersion trend of σ . The second pixel in the random pair is a pixel extracted from the Gaussian distribution centered on the first pixel, with a standard deviation of $\sigma/2$. Experience shows that such Gaussian selection can improve the feature point matching rate.

(4) Construct a binary descriptor by comparing the brightness of two pixels in a random pair as key points. If the first pixel is brighter than the second, the corresponding bit in the descriptor is assigned 1, otherwise it is assigned 0. The first bit of the feature vector corresponds to the first random point pair of this key point, then BRIEF selects a new random pixel pair for the same key point, compares their brightness and assigns 1 or 0 to the next bit in the feature vector. Assuming a smooth image, the τ test is performed in the field P of size $S \times S$, then the binary formula for grayscale difference can be expressed as:

$$\tau(p; x, y) = \begin{cases} 1: I(p, x) < I(p, y) \\ 0: I(p, x) \geq I(p, y) \end{cases} \quad (6)$$

Among them, $I(p, x)$ is the pixel gray value of the smoothed image neighborhood P at point $x = (u, v)^T$.

(5) BRIEF repeats steps (3) (4) N times for the same key point, uses formula (6) to obtain the response value, and puts the comparison results of the brightness of N pixels into the binary feature vector of the key point to form a feature point. And then turn to the next key point until the descriptor is created for each key point in the image. The generated binary descriptor can be expressed as:

$$f_n(p) = \sum_{1 \leq i \leq n} 2^{i-1} \tau(p; x_i, y_i) \quad (7)$$

Among them, N can be 128, 256 and 512 bits, and the comprehensive generation speed and the distribution and accuracy of descriptors, 256-bit descriptors are the most advantageous.

The BRIEF algorithm does not have rotation invariance and scale invariance, and is more sensitive to noise. The ORB algorithm improves the BRIEF algorithm in two ways:

(1) The main direction of the feature points obtained by the improved FAST algorithm, first rotate the descriptor generated by BRIEF, and then discriminate and binary encode, so that the descriptor has rotation invariance, called steered BRIEF descriptor.

The ORB algorithm uses the main direction of the key point to rotate the BRIEF descriptor. For any feature point, the set of the location of (x_i, y_i) n pairs of points in the neighborhood of 31×31 can be represented by a $2 \times n$ matrix:

$$S = \begin{bmatrix} x_1, \dots, x_n \\ y_1, \dots, y_n \end{bmatrix} \quad (8)$$

Using the main direction θ of the feature point and the corresponding rotation matrix R_θ obtained by FAST, the rotation S_θ can be obtained to represent S :

$$R_\theta = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (9)$$

$$S_\theta = R_\theta S \quad (10)$$

Therefore, the final rotating BRIEF descriptor can be expressed as:

$$g_n(p, \theta) := f_n(p) | (x_i, y_i) \in S_\theta \quad (11)$$

(2) The integrated image is used to solve the noise sensitivity of the BRIEF algorithm and enhance

the anti-noise ability of the descriptor.

The brief descriptor has an average value of about 0.5 and a variance of 0.25, which is very distinguishable. Although the improved rotation BRIEF has rotation invariance, the mean is more discrete, the variance is unevenly distributed, and the distinguishability is lower. The ORB algorithm uses statistical learning Algorithm, recalculate the set of binary test point pairs, the steps are as follows:

Within the 31×31 neighborhood of feature points, random point pairs are generated, with the random point as the center, and a sub-window of 5×5 is taken. The average gray value of the sub-window is used to replace the original single pixel value for comparison, and $N = (31-5+1)(31-5+1) = 729$ patches that can be compared, there are $M = (C, 2)$ point pairs, that is, M 0,1 character strings, M is much greater than 256, According to the principle of maximizing the variance between the mean and the points, the greedy algorithm is used to screen out 256. If the screening is less than 256, the threshold is lowered until 256 is satisfied.

This algorithm is a greedy search for an uncorrelated test whose mean is close to 0.5, which is called rBRIEF. The rBRIEF descriptor is significantly better than steered BRIEF in variance and correlation, with better discrimination and low dimensional correlation.

3. Image feature point matching

When the default matching algorithm of the ORB algorithm is used, there are many wrong matches and the matching accuracy is not high. You can use the RANSAC algorithm to filter the wrong matches and improve the matching accuracy. The RANSAC algorithm is the abbreviation of RANdom SAMple Consensus, which means that the random sampling is consistent. It is based on a set of sample data sets containing abnormal data (also known as outliers), and through multiple iterations, the mathematical model parameters of the data are calculated to obtain valid sample data (also known as inside points). Is used to find an optimal homography matrix H . It was first proposed by Fischler and Bolles in 1981. The size of matrix H is 3×3 , which can be expressed as:

$$s \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (12)$$

(x, y) is the feature point position of the target image, (x', y') is the feature point position of the scene image, and s is the scale parameter.

The purpose of the RANSAC algorithm is to find the optimal parameter matrix so that the number of data points satisfying the matrix is the most, usually let $h_{33} = 1$ normalize matrix. This matrix has 8 unknown parameters, so at least 8 linear equations need to be solved. Corresponding to the point position information, a set of point pairs can list two equations, then at least 4 sets of matching point pairs are required. The RANSAC algorithm randomly draws 4 samples from the matching data set and ensures that these 4 samples are not collinear, calculates the homography matrix, and then uses this model to test all the data, and calculates the number of data points and projections error (ie. cost function) that satisfy this model. If this model is the optimal model, the corresponding cost function is the smallest. The cost function can be calculated using the following formula:

$$\sum_{i=1}^n \left(x'_i - \frac{h_{11}x_i + h_{12}y_i + h_{13}}{h_{31}x_i + h_{32}y_i + h_{33}} \right)^2 + \left(y'_i - \frac{h_{21}x_i + h_{22}y_i + h_{23}}{h_{31}x_i + h_{32}y_i + h_{33}} \right)^2 \quad (13)$$

The specific steps of the algorithm are as follows:

- (1) Randomly extract 4 sample data from the data set (the four samples cannot be collinear), calculate the transformation matrix H , and record it as model M ;
- (2) Calculate the projection error of all data in the data set and the model M . If the error is less than the threshold t , add the interior point set I ;
- (3) If the number of elements in the current interior point set I is greater than the optimal interior

point set I_{best} number d , update $I_{best}=I$ and update the number of iterations k at the same time;

(4) If the number of iterations is greater than k , then exit; otherwise, increase the number of iterations by 1 and repeat the above steps.

The parameters t and d are determined through experiments according to specific problems and data sets. The number of iterations k is continuously updated when it is not greater than the maximum number of iterations. It can be inferred from the theoretical results and can be calculated by the following formula:

$$k = \frac{\log(1-p)}{\log(1-w^n)} \quad (14)$$

Among them, p is the confidence level, usually 0.995, w is the probability ratio of selecting a valid sample from the data set each time.

After obtaining the homography matrix, retain the valid points after filtering, and discard the invalid points, thus completing the filtering of false matches.

4. Experimental results and analysis

In order to truly embody the traditional ORB and RANSAC combination algorithm in various types of images, the matching degree of the ORB algorithm is higher and shorter, and this experiment uses the Mikolajczyk standard image library in the Oxford classic image matching database, which contains 8 sets of image sequences. Each group of 6 images with different changes. In the image data set, the Wall and Graffiti image groups are the angle change sequences, the Boat and Bark image groups are the scale and rotation change sequences, the Bikes and Trees image groups are the blur degree change sequences, and the Leuven image group is the brightness change sequence [12]. We select the first and second images in each group to form an experimental group. The platform is Spyder, the programming language is Python3.7, the computer processor is Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz 2.20 GHz CPU, and the memory is 64.0 GB.

The experiment compares the traditional ORB algorithm and the algorithm combining ORB and RANSAC, and compares the pros and cons of the algorithm through the matching accuracy of the feature points and the matching time-consuming, as shown in the table, and is visually represented by the line chart, as shown in the figure.

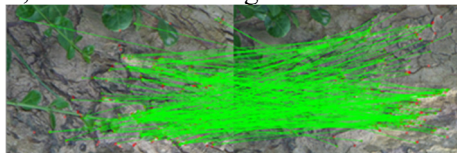


Figure 1. Bark (ORB)

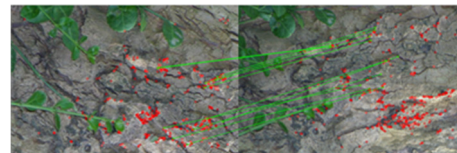


Figure 2. Bark (ORB+RANSAC)



Figure 3. Bikes (ORB)



Figure 4. Bikes (ORB+RANSAC)

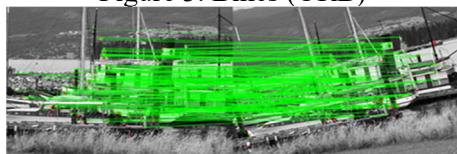


Figure 5. Boat (ORB)

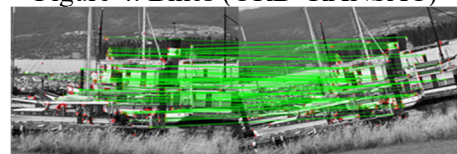


Figure 6. Boat (ORB+RANSAC)



Figure 7. Graffiti (ORB)



Figure 8. Graffiti (ORB+RANSAC)



Figure 9. Leuven (ORB)



Figure 10. Leuven (ORB+RANSAC)



Figure 11. Trees (ORB)



Figure 12. Trees (ORB+RANSAC)

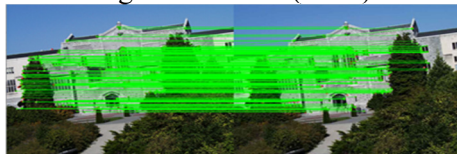


Figure 13. Ubc (ORB)

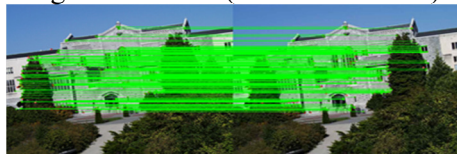


Figure 14. Ubc (ORB+RANSAC)

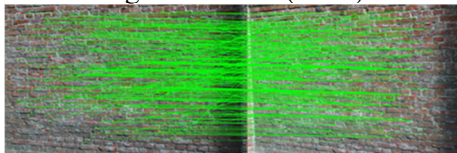


Figure 15. Wall (ORB)

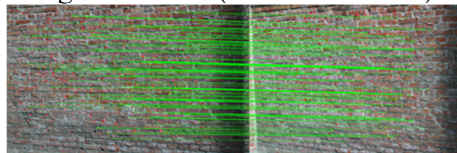


Figure 16. Wall (ORB+RANSAC)

Table 1. Comparison of matching accuracy of feature points.

Algorithm	Matches/Correct Numbers								Accuracy %
	Bark	Bikes	Boat	Graffiti	Leuven	Trees	Ubc	Wall	
ORB	253/21	343/163	311/141	295/138	301/176	260/90	427/408	300/122	47.04
ORB+RANSAC	15/10	272/168	154/112	145/91	220/152	120/69	417/400	124/92	70.08

Table 2. Time-consuming comparison of feature point matching.

Algorithm	Time Consuming (ms)								Average time
	Bark	Bikes	Boat	Graffiti	Leuven	Trees	Ubc	Wall	
ORB	117.93	86.98	79.95	66.96	67.97	107.99	74.96	80.92	85.46
ORB+RANSAC	43.97	77.99	69.96	57.97	63.96	90.94	66.92	77.97	68.71

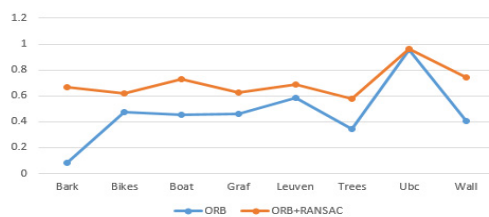


Figure 17. Comparison of accuracy of feature point matching

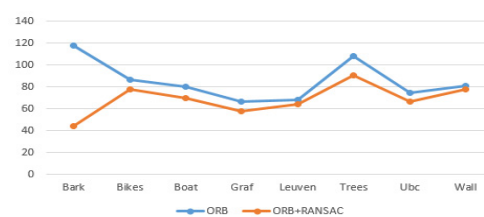


Figure 18. Comparison of time-consuming feature point matching

From the above chart, it can be seen that compared with the ORB algorithm, the matching accuracy of the algorithm combined by ORB and RANSAC is improved by 22.96%, and the time consumption is reduced by 24.38%.

5. Conclusion

This paper analyzes the realization process of traditional ORB and ORB combined with RANSAC two algorithms to achieve feature point matching, and through experiments using the Mikolajczyk standard image library in the Oxford classic image matching database as the data set, it is verified that the

algorithm combined with RANSAC is better than the traditional matching algorithm. Both accuracy and time-consuming are far superior to the traditional ORB algorithm.

Acknowledgments

This research was jointly supported by the National Natural Science Foundation of China (41971193; 41871236); Foundation of Jilin Provincial Science & Technology Department (20190201265JC; 20180622006JC; 20200301045RQ); Foundation of Jilin Province Education Department (JJKH20191260KJ).

References

- [1] Uchiyama, Y., Abe, A., Muramatsu, C., et al. (2015) Eigenspace template matching for detection of lacunar infarcts on MR images. *Journal of Digital Imaging*, 71(1) : 85
- [2] Reese, H., Nordkvist, K., Nystrom, M., et al. (2015) Combining point clouds from image matching with SPOT 5 multispectral data for mountain vegetation classification. *International Journal of Remote Sensing*, 36(2) :403-416.
- [3] Ye, Y., Li, S., Ming, H., et al. (2017) Robust Optical - to- SAR Image Matching Based on Shape Properties . *IEEE Geoscience & Remote Sensing Letters*, (99) :1 -5.
- [4] Abate, A.F, Nappi, M., Riccio, D., et al. (2007) 2D and 3D face recognition: A survey. *Pattern Recognition Letters*, 28 (14) :1885- 1906.
- [5] Mur-Artal, R., Miel, J.M.M., Tardos, J.D. (2015) ORB-SLAM: A versatile and accurate monocular SLAM system .*IEEE Transactions on Robotics*, 31 (5) :11474163.
- [6] Zhang, T., Liu, J.H., Liu, S.L., et al. (2017) A 3D reconstruction method for pipeline inspection based on multivision . *Measurement*, 98:35 -48.
- [7] Lowe, D.G. (2004) Distinctive image features from scale invariant keypoints . *International Journal of Computer Vision*, 60(2) :91410.
- [8] Rublee, E., Rabaud, V., Konolige K, et al. (2012) ORB: An efficient alternative to SIFT or SURF. In: 2011 International Conference on Computer Vision. Barcelona. IEEE.
- [9] Rosten, E., Porter, R., Drummond, T. (2009) Faster and better: A machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1) :105-119.
- [10] Calonder, M., Lepetit, V., Stretha, C., et al. (2010) BRIEF: Binary robust independent elementary features. In: 2010 European Conf on Computer Vision. Heraklion. pp. 778 – 792.
- [11] Xing, K.S., Ling, Y.Z., Chen, M.Y. (2016) Research on erroneous matching point elimination algorithm for ORB feature matching. *Journal of Electronic Measurement and Instrument*, 30(08):1255-1262.
- [12] Mikolajczyk, K., Schmid, C. (2005) A Performance Evaluation of Local Descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1615-30.