

# Research on the algorithm of image feature detection and matching

**Jiali Zhang**

College of Landscape Achitecture, NanJing Forestry University, Nanjing, China

ma.tianyu@northeastern.edu

**Abstract.** In-depth research on feature detection technology affects people's modern life. Modern artificial intelligence can act as the eyes of human beings and efficiently filter out effective information from complex pictures. Corner detection has now evolved into a tool for efficient image scanning. People's increasingly stringent requirements for image processing continue to promote the birth of new technologies. Corner detection methods have been improved and perfected, and have experienced detectors such as Harris, FAST, Scalriant Feature Transform (SIFT), Speeded Up Robust Feature (SURF), Binary Robust Invariant Scalable Keypoints (BRISK), KAZE, AKAZE and Oriented FAST and Rotated BRIEF (ORB). In the face of many mainstream detectors, the main research purpose of this paper is to rely on Python and Computer vision to study the detection efficiency of various detectors in different environments. In this experiment, thirteen groups of pictures were selected, and after being flipped, complicated, and blurred respectively, they were detected by different detectors to obtain the results. Finally, by comparing the feature detection points, detection time and other factors, this study found that the ORB detector can be competent in most situations and is currently the fastest and stable feature point detection and extraction algorithm. On the other hand, the BRISK detector can handle highly blurred images more efficiently.

**Keywords:** Computer vision; feature detection and matching.

## 1. Introduction

With the rapid progress of computer technology and the further development of artificial intelligence, deep learning technology has been introduced into VR and achieved very good results. As an important branch of computer science and technology, computer vision has been devoted to solving graphics problems with algorithms. Much of this scientific achievement revolves around how the human eye works. Feature detection and matching are used in almost all computer vision tasks. A typical example is face recognition. This process requires artificial intelligence to be mainly used for security inspection, identity verification and mobile payment when there is a lot of information; commodity identification is mainly used in the process of commodity circulation, especially in unmanned retail areas such as unmanned shelves and smart retail cabinets. In this process, artificial intelligence imitates the structure of the human eye, captures actual image information, and then passes through complex internal conversions, and finally outputs valid information. Image recognition refers to the use of computers to process, analyse and understand images to identify objects and various patterns of objects. It is a practical application of image feature detection and includes image stitching, camera calibration, dense

reconstruction (indirectly using feature points as seed points to diffuse and match to obtain dense point clouds), scene understanding (bag of words method, feature points as the centre to generate key features for scene identification) and so on.

The main task of this paper is to apply feature detection and matching techniques to compare images for similarity. Also, the main method used in this study is point features. First, a corner is defined as the intersection of two or more sides. More precisely, a corner point is a point with a sufficiently high grayscale value in all directions in the neighbourhood and is a point with a maximum curvature on the edge curve of the image. Secondly, there are two basic requirements for image feature points, difference, and repeatability. Differences are detectable from salient points in the visual scene, including where the grayscale changes significantly. Repetition means that the same feature is repeated in different perspectives, and it is matchable. This experiment is based on detectability, and the purpose is to determine the matching of the pictures.

## 2. Related Work

Several traditional feature point detection methods are Harris, SIFT and SURF. The Harris corner detection operator can not only detect the grayscale differences in four horizontal and vertical directions, but also detect the grayscale differences in various directions, and has rotation invariance and stability for partial affine transformations. However, the limitation of the Harris operator is that it is scale-sensitive and not scale-invariant [1]. Specifically, the larger the magnification, the more difficult the detection, which is caused by too few feature points. In response to this improvement, the SIFT operator was discovered, which is a scale space-based image local feature description operator that remains invariant to graph scaling, rotation, and even affine transformations. The specific steps of point detection in the SIFT algorithm are to construct a scale space, construct a Gaussian difference scale space, detect extreme points in the DOG scale space, accurately locate feature points, and remove unstable points [2]. SURF features are further optimizations to Harris features. It constructs a pyramid scale space based on a Hessian matrix and uses a box filter to simplify 2D Gaussian filtering without down sampling [3].

In addition, some more excellent feature detection algorithms have also been developed in modern times, such as BRIEF, KAZE, AKAZE and ORB. BRISK is a binary feature descriptor. Its biggest advantage is that it can effectively handle images with a large blur. KAZE is a process for nonlinear diffusion processing in the image domain. Its biggest advantage is that the processing speed is accelerated, and the problem of border blurring is solved. AKAZE is also a local feature descriptor, which can be regarded as an improvement based on KAZE. The biggest feature is that it has rotational invariance [4]. ORB is a research result in recent years, it is a fast feature point extraction and description algorithm. The ORB feature combines the detection method of FAST feature points with the Brief feature descriptor and improves and optimizes them based on their original ones. According to Alex's knowledge in his blog, the eigenvectors created by the ORB algorithm contain only 1s and 0s and are called binary eigenvectors. This algorithm greatly improves the analysis efficiency, it is about 100 times faster than SIFT and 10 times faster than SURF. This study selected the most representative SIFT, BRIEF, KAZE, AKAZE and ORB five algorithms for comparison, and choose the best way to extract features for the target image.

## 3. Method

### 3.1. Problem Description

Artificial intelligence has been widely used in target detection and recognition. The most common is the stitching of pictures or videos, and the target objects are screened out from pictures or videos. In today's highly data-driven world, people need tools that run faster and have more accurate recognition capabilities to help us accomplish such goals. This report focuses on the feature detection efficiency of images and the explanation of algorithm principles. The main analysis systems used are Python and OpenCV.

### 3.2. Data Collection

The raw data for this report comes from a set of original life photos. Its main content is a bunch of messy and disordered objects. Our goal is to choose the most efficient and accurate algorithm to find the target object from many objects. Its core idea is feature matching. The software and platforms used in this project are Python (version 3.10.4) and OpenCV (version 3.3.1), and feature detection and matching are performed on SIFT, ORB, KAZE, and AKAZE through the BF (Brute Force) algorithm.

### 3.3. Technical Approach

**3.3.1. SIFT Detector.** SIFT features [5] are very stable image features. The SIFT algorithm covers the essential ideas of image feature extraction, from the detection of feature points to the generation of descriptors, to complete the accurate description of the image. It applies a Gaussian kernel function. The Gaussian kernel function is the only scale-invariant kernel function. Using Gaussian filtering on the image can blur the image. Using different "Gaussian kernels" can get images with different degrees of blurring, which can simulate the distance from the target to the target. Formation process on the retina. Each feature point is described by a 128-dimensional vector, and each dimension occupies 4 bytes. SIFT requires  $128 \times 4 = 512$  bytes of memory [6]. The relevant keywords are described in Table 1.

**Table 1.** SIFT detector algorithm.

Algorithm 1: SIFT detector algorithm	
1.	Generate Gaussian Difference Pyramid (DOG Pyramid), scale space construction
2.	Spatial extreme point detection (preliminary exploration of key points)
3.	Precise positioning of stable key points
4.	Stable key point orientation information allocation
5.	Description of key points
6.	Feature point matching

**3.3.2. ORB Detector.** The biggest disadvantage of traditional feature point descriptors such as SIFT is that it runs slowly and occupies a large space. ORB (Oriented Brief) can solve such a problem very well (Table 2). In the ORB scheme, FAST is used as the feature point detection operator first, and then the improved Brief is used to calculate the descriptor [7]. An important contribution of ORB is the introduction of directions for locating key points [8].

**Table 2.** ORB detector algorithm.

Algorithm 2: ORB detector algorithm	
1.	ORB chose Brief as the feature description method
2.	One of the surprising features of BRIEF is that: for each feature bit of an n-dimensional binary string, the value of all feature points at this bit satisfies a Gaussian distribution with a mean close to 0.5 and a large variance This forms the vector T. The larger the variance, the stronger the discrimination, and the greater the difference between the descriptors of different feature points, which is not easy to be mismatched for matching.
3.	Greedy search:
a)	First, put the first test value into R, so it can be eliminated one by one in the vector T.
b)	Then get the next test value from the vector T and compare it with all the values in R; discard it if it exceeds the expected threshold, otherwise put it in R.
c)	Repeat the action and guide until there are 256 column vectors in R. If all the values in the vector T are found, and there are not enough 256, then increase the threshold and repeat the operation.

**3.3.3. KAZE Detector.** SIFT, SURF and other features are all detected by a Gaussian kernel in a linear scale space. The transformation of each point at the same scale is the same. Since the Gaussian function

is a low-pass filter function, it will smooth the edges of the image, so that the image is lost. Lots of details. The Gaussian pyramid established by KAZE is a nonlinear scale space, and the additive operator splitting algorithm (AOS) is used for nonlinear diffusion filtering. A notable feature is that it preserves edge details while blurring images. In addition, KAZE has greatly increased the operating speed [9].

**3.3.4. AKAZE Detector.** KAZE is a feature point detection and description algorithm newly proposed by ECCV in 2012, and AKAZE is improved based on KAZE. It can be seen as an accelerated version of KAZE (Table 3).

**Table 3.** AKAZE detector algorithm.

Algorithm 3: AKAZE detector algorithm
1. Nonlinear diffusion filtering and scale space construction: (Perona-Malik diffusion equation) → (AOS algorithm)
2. Hessian matrix feature point detection
3. Feature detection and descriptor generation: (main direction of feature points: first-order differential image) → (construct feature description vector)

**3.3.5. BRISK Detector.** As we know, in order to speed up the computation efficiency, keypoints can be detected in the layers or between layers of the pyramid image. The location and scale of each keypoint are obtained in the continuous region by means of quadratic function fitting. The BRISK algorithm is also a feature extraction algorithm, which is a binary feature description operator [10]. In the BRISK algorithm, an image pyramid is constructed for multi-scale expression to solve the problem of scale invariance. In addition, BRISK also has an excellent ability to detect blurred images, because in BRISK (Table 4), its 16-pixel circle provides at least 9 consecutive pixels, so that no matter if it is brighter or darker than the center pixel, in the case of it can be successfully executed under the standard of FAST.

**Table 4.** BRISK detector algorithm.

Algorithm 4: BRISK detector algorithm
1. The image pyramid is constructed in the BRISK algorithm for multi-scale expression, so it has better rotation invariance, scale invariance, and better robustness. It has better rotation invariance, scale invariance, better robustness, etc.
2. When registering images with large blur, the BRISK algorithm performs best.

## 4. Experimental Procedure

### 4.1. Image Selection

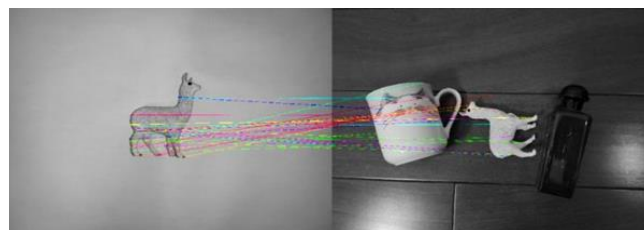
In this project, this paper's data source is a large range and a small range of photos taken in life. This study expects the target object to be detected from the features in the overall photo. The experiments used SIFT detector, ORB detector, KAZE detector AKAZE detector and BRISK detector for evaluation. The evaluation metrics are the accuracy and computation time of the original image. It should be noted that this experiment tested ten groups of pictures, and in this report, only four groups of pictures were chosen to be displayed and the experimental data of the first group was introduced. The image to be recognized is image1, the image rotated and placed in a complex environment is image2, the high-definition image in the same environment is image3, and the highly blurred image is image 4, as figure 1.



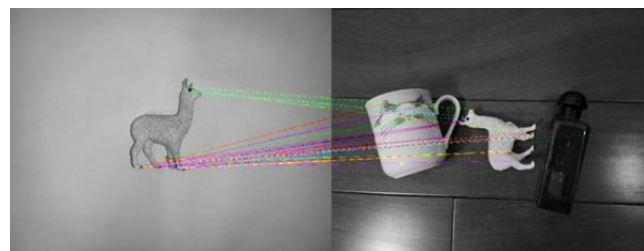
**Fig. 1.** Four sets of original images. Each set, from left to right are: image 1, image 2, image 3 and image 4.

#### 4.2. Results

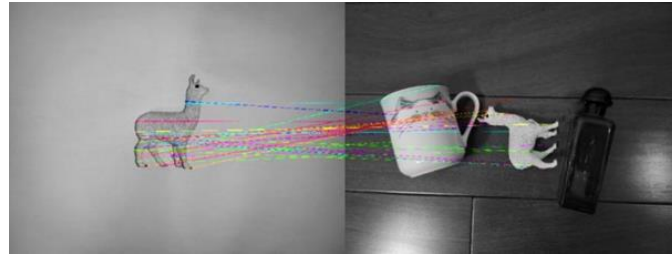
- About the comparison of image1 and flip result image2 (figure 2)



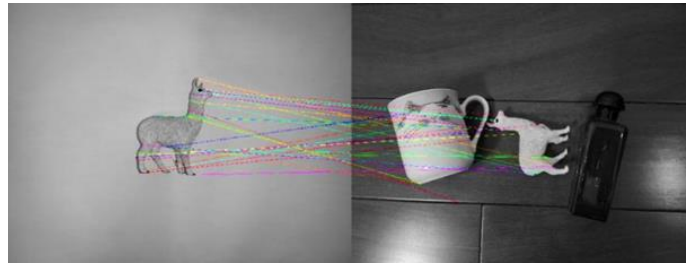
(a)



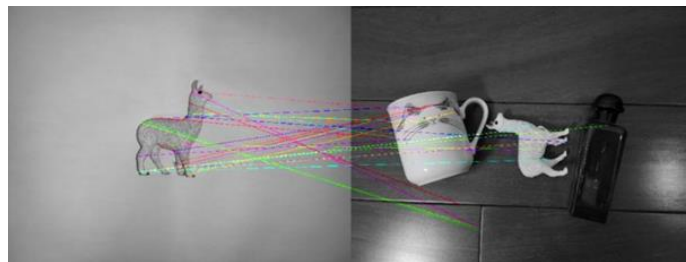
(b)



(c)



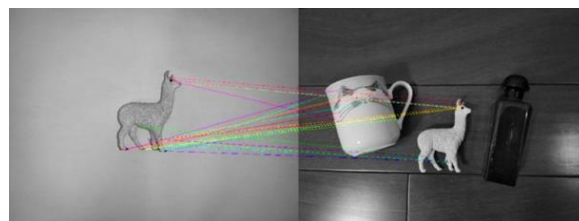
(d)



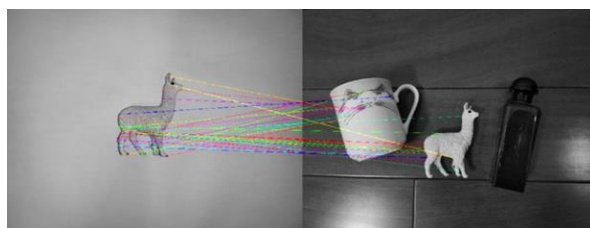
(e)

**Fig. 2.** Comparison of image 1 and 2. From top to bottom are: results of the SIFT, ORB, KAZE, AKAZE and BRISK detectors.

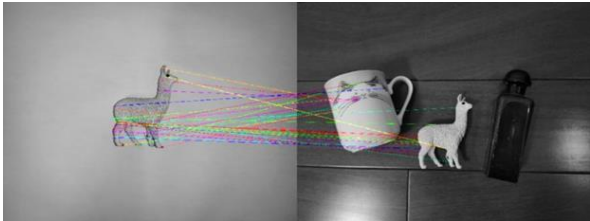
● *About the comparison of image1 and high definition image3 in complex environments (figure 3)*



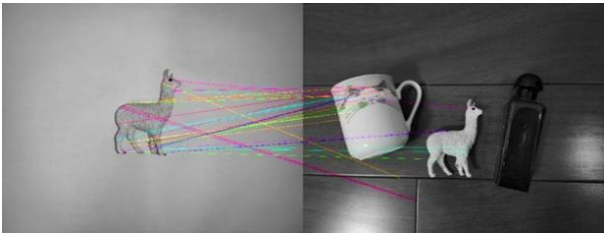
(a)



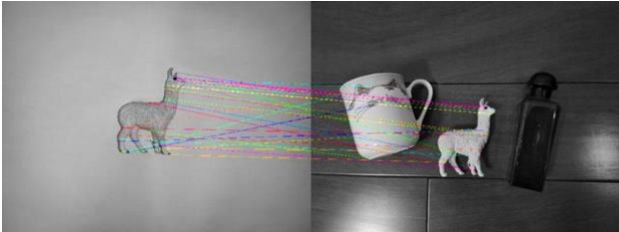
(b)



(c)



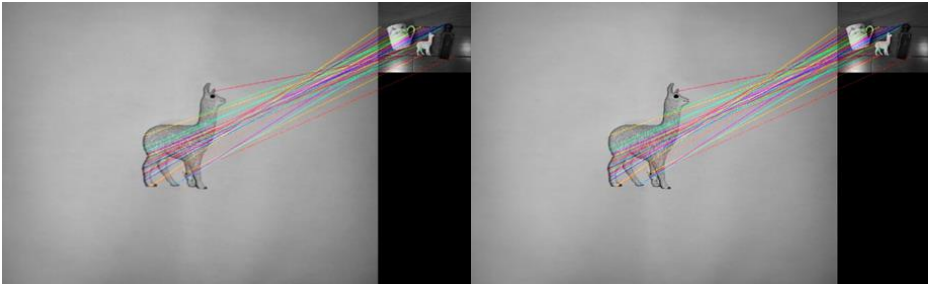
(d)



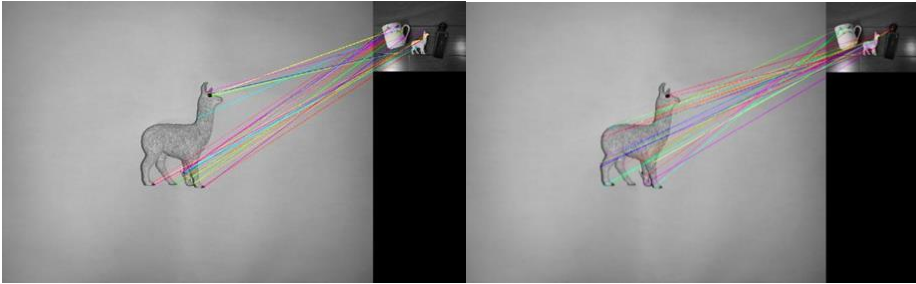
(e)

**Fig. 3.** Comparison of image 1 and 3. From top to bottom are: results of the SIFT, ORB, KAZE, AKAZE and BRISK detectors.

- *About the comparison of comparing image1 and highly blurred image4(figure 4)*

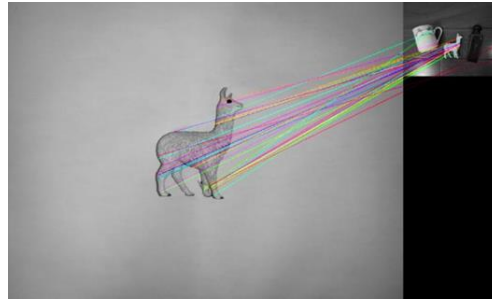


(a)



(b)





(c)

**Figure 4.** Comparison of image 1 and 4. From top to bottom are: results of the SIFT, ORB, KAZE, AKAZE and BRISK detectors.

#### 4.3. Formation of All Data

Finally, all three sets of valid results shown in Visual Studio Code are listed in the Table 5 -Table 7 for easy observation and comparison.

**Table 5.** The comparison results of image 1 and image 2.

Detector	Size of Descriptor	Computation time	Key points 1	Key points 2	Match rate
SIFT	128	11.2935	4383	11939	0.313
ORB	32	5.8878	500	500	0.645
KAZE	128	11.5466	4383	11939	0.488
AKAZE	61	9.5662	1802	3194	0.497
BRISK	64	5.8275	2215	3366	0.286

First, in the first scenario, the matching speed of the detector in the flip case with higher pixels is as follows: ORB > BRISK > AKAZE > SIFT > KAZE. At the same time, the order of matching rate is: ORB > AKAZE > KAZE > SIFT > BRISK. It can be seen from this experiment that in the case of deformation, the ORB detector is not only time-consuming, but also has high feature matching efficiency, making it the first choice for this scene. At the same time, AKAZE, as an accelerated version of KAZE, does perform better than the KAZE detector. This is mainly reflected in its ability to match more information in less time.

**Table 6.** The comparison results of image 1 and image 3.

Detector	Size of Descriptor	Computation time	Key points 1	Key points 2	Match rate
SIFT	128	10.5502	4383	7251	0.231
ORB	32	5.7959	500	500	0.366
KAZE	128	10.8033	4383	7251	0.409
AKAZE	61	9.5988	1802	3341	0.382
BRISK	64	13.0894	2215	23010	0.310

Secondly, in the second scenario, this study sets up the flip of the graph. The ORB detector still maintains efficient performance, and can complete feature matching in 5.8 seconds, which is the fastest and best among the five detectors. However, KAZE detects more feature points in a limited time. In addition, the matching ratio of each detector in the flip case is: KAZE > AKAZE > ORB > BRISK > SIFT. All things considered, KAZE performs better in this flip scenario.



**Table 7.** The comparison results of image 1 and image 4.

Detector	Size of Descriptor	Computation time	Key points 1	Key points 2	Match rate
SIFT	128	6.3033	4383	708	0.08
ORB	32	3.4838	500	500	0.245
KAZE	128	5.8392	4383	708	0.18
AKAZE	61	5.2963	1802	555	0.13
BRISK	64	3.2200	2215	1232	0.275

Finally, set a highly blurred scene, the original image pixel is 4000\*3000, and the processed image4 pixel is 1000\*750. In this case, the matching rate of each detector has dropped, and the ranking is as follows: BRISK > ORB > KAZE > AKAZE > SIFT. However, BRISK and ORB perform well, with more detection points falling on the target. Look again at the test speed data: BRISK > ORB > AKAZE > KAZE > SIFT. It can be found that BRISK is in a leading position in terms of match rate and speed. Therefore, in this experiment, the BRISK detector performs best when the target is highly blurred.

## 5. Conclusion

In this project, a target photo image1 was placed it in different situations to measure match speed and match rate of different detectors in different situations. The three scenarios are deformation (rotation occurs), complex environment, and high blur. Throughout the entire testing process, this study believes that the ORB detector is competent in most cases and is currently the fastest and most stable feature point detection and extraction algorithm. On the other hand, KAZE can correctly detect more feature points in the flip case, but it takes more time. Such properties of it can be useful in special cases. Last but not least, the BRISK detector can handle highly blurred images more efficiently. In future research, the use of feature detectors to stitch two photos will be considered. In addition, the technology of portrait detection will become the next stage of learning tasks, that is, the application of AR feature detection in video recognition.

## References

- [1] Senit\_Co. (2017). Harris corner detection of image features. *Senitco.github.io*. <https://senitco.github.io/2017/06/18/image-feature-harris/>
- [2] CharlesWu123. (2021). OpenCV —— SIFT feature detector for feature point detection. *CSDN*. [https://blog.csdn.net/m0\\_38007695/article/details/115524748](https://blog.csdn.net/m0_38007695/article/details/115524748)
- [3] Bansal, Kumar, M., & Kumar, M. (2021). 2D object recognition: a comparative analysis of SIFT, SURF and ORB feature descriptors. *Multi. Tools Appl.*, **80(12)**, 18839–18857.
- [4] SongpingWang. (2021). OpenCV + CPP Series (22) Image Feature Matching (KAZE/AKAZE). *Electro. Des. Eng.*, **438**
- [5] Dellinger, Delon, J., Gousseau, Y., Michel, J., & Tupin, F. (2015). SAR-SIFT: A SIFT-Like Algorithm for SAR Images. *IEEE Trans. Geos. Re. Sens.*, **53(1)**, 453–466.
- [6] OpenCV school. (2019). Detailed explanation and use of OpenCV SIFT feature algorithm. *cloud.tencent*. <https://cloud.tencent.com/developer/article/1419617>
- [7] Alex777. (2019). ORB Feature Extraction Algorithm (Theory). *Cnblogs*. <https://www.cnblogs.com/alexme/p/11345701.html>
- [8] Matusiak, Skulimowski, P., & Strumillo, P. (2017). Unbiased evaluation of keypoint detectors with respect to rotation invariance. *IET Comput. Vis.*, **11(7)**, 507–516.
- [9] SongpingWang. (2021). OpenCV + CPP Series (22) Image Feature Matching (KAZE/AKAZE). *CSDN*. [https://blog.csdn.net/wsp\\_1138886114/article/details/119772358](https://blog.csdn.net/wsp_1138886114/article/details/119772358)
- [10] Hujingshuang. (2015). BRISK Feature Extraction Algorithm. *CSDN*. <https://blog.csdn.net/hujingshuang/article/details/47045497>