International Journal of Electrical and Electronic Engineering & Telecommunications

*ISSN 2319 – 2518 www.ijeetc.com*
*Vol. 6, No. 2, April 2017*


*Research Paper*

# A SURVEY OF TRENDS IN LOCAL INVARIANT FEATURE DETECTORS

**J Naga Surekha[1]*, Y Sri Chakrapani[2] and N.Venkatewsara Rao[3]**

*Corresponding Author: **J Naga Surekha,** ✉jonnalasurekha666@gmail.com

In this paper, the synopsis of local invariant interest point detectors, their working, advantages, disadvantages and recent trends are presented. The features detector extracts the features from the image, e.g., a corner, blob or edge detector. A features detector is said to be invariant if under a certain family of transformations if its value does not change when a transformation from this family is applied to its argument. The characteristics of this detector are robustness, repeatability, accuracy, generality, efficiency, quantity etc. Some of the feature detectors are SIFT SURF, FAST, BRISK, ORB and HCD. These descriptors are compared according to their average processing time per frame.

***Keywords:*** Feature, SURF, FAST, ORB, HCD

## INTRODUCTION

Feature detection is a low-level image processing operation [4]. It is usually performed as the first operation on an image, and examines every pixel to see if there is a feature present at that pixel. If this is a part of larger algorithm, then the algorithm will typically examine the image in the region of the features. As a built-in pre-requisite to feature detection, the input image is usually smoothed by a Gaussian kernel in a scale space representation and one or several feature images are computed, often expressed in terms of local image derivatives operations.

Occasionally, when feature detection is computationally expensive and there are time constraints, a higher level algorithm may be used to guide the feature detection stage, so that only certain parts of the image are searched for features.

1 M. Tech Student, Embedded Systems, Gudlavalleru Engineering College, Gudlavalleru, India.
2 Assoc Prof., Department of ECE, Gudlavalleru Engineering College, Gudlavalleru, India.
3 Professor, Department of ECE, Bapatla Engineering College, Bapatla , India.

Many computer vision algorithms use feature detection as the initial step, so as a result, a very large number of feature detectors have been developed. These vary widely in the kinds of feature detected, pedestrian detection (Gavrila, 2000) the computational complexity and the repeatability.

## LOCAL INVARIANT FEATURES

A local feature is an image pattern which differs from its immediate neighborhood. It is usually associated in (Belongie *et al*., 2002) with a change of an image property or several properties simultaneously, although it is not necessarily localized exactly on this change. The image properties commonly considered are intensity, color, and texture.

Local invariant features are a powerful tool that has been applied successfully in a wide range of systems and applications. In the following, we distinguish three broad categories of feature detectors based on their possible usage. It is not exhaustive or the only way of categorizing the detectors but it emphasizes different properties required by the usage scenarios.

First, one might be interested in a specific type of local features in Gevers T and (Smeulders, 1999), as they may have a specific semantic interpretation in the limited context of a certain application. For instance, edges detected in aerial images often correspond to roads; blob detection can be used to identify impurities in some inspection task; etc. These were the first applications for which local feature detectors have been proposed. Second, one might be interested in local features since they provide a limited set of well localized and individually identifiable anchor points.

## CHARACTERISTICS OF FEATURE DETECTORS

D G Lowe *et al*. defined a local feature as "*an image pattern which differs from its immediate neighborhood*" (Lowe, 2004). The purpose of local invariant features is to provide a representation that efficiently matches local structures between images. That is, a sparse set of local measurements will be obtained that capture the essence of the underlying input images and encode their interesting structures. To meet this goal, the feature detectors and extractors must have certain properties keeping in mind that the importance of these properties depends on the actual application settings and compromises need to be made. The following properties are important for utilizing a feature detector in computer vision applications:

- Robustness, the feature detection algorithm (Broggi *et al*., 2005) should be able to detect the same feature locations independent of scaling, rotation, shifting, photometric deformations, compression artifacts, and noise.
- Repeatability, the feature detection algorithm should be able to detect the same features of the same scene or object repeatedly under variety of viewing conditions.
- Accuracy, the feature detection algorithm should accurately localize the image features (same pixel locations), especially for image matching tasks, where precise correspondences are needed to estimate the epi-polar geometry.
- Generality, the feature detection algorithm should be able to detect features that can be used in different applications.

- Efficiency, the feature detection algorithm should be able to detect features in new images quickly to support real-time applications.
- Quantity, the feature detection algorithm in (Lowe, 2004) should be able to detect all or most of the features in the image. Where, the density of detected features should reflect the information content of the image for providing a compact image representation.

## TYPES OF FEATURE DETECTORS

The several types of feature detectors are given below:

**HCD:** The traditional Harris corner detection algorithm is sensitive to scale change, corners detected throughout the entire image under complex background, thus extracting more false corners, lead to the follow-up of large amount of calculation and a high rate of error matching. To solve this problem, an optimized Harris corner detection algorithm. First, a significant region detection method is used to extract the target area, and take closing operation for the result figure, can effectively achieve target and background segmentation; second, scale invariant describing methods is applied to Harris algorithm, suppression methods to extract corners, get more right corners.

At the same time, combined with the non-maximum suppression methods to extract corners, get more right corners.

According to above problem of Harris, the algorithm can be divided into two parts: The first part, saliency detection method is used to extract the target in the image, corners are more concentrated in the target area. The second part: The method of scale space theory combing with the non-maximum suppression is used to detect corner, can improve the automation capabilities of the image.

The image information can be divided into two parts: redundancy and change. Since the human visual is sensitive to the change of the region, significant region detection retains conversion section of image, and removes redundant parts. Literature (Gevers and Smeulders, 1999) and (Harris and Stephens, 1998) proposed a simple calculation model based on image visual significance, by calculating the residual spectrum of logarithm to extract significant region.

There still may be many scattered, small salient region on the background part of the image to interfere with selecting for target part, to make the generated target figure not enough clear. Thus, after threshold segmentation, take the operation from corrosion to expansion for the binary figure, eliminate the small salient regions on the background part and fill small voids on the target part, to connect adjacent objects and smooth boundary. We use closing operation to erosion and dilation image.

Scale space describes the image characteristics at different scales. Lindeberg pointed out that the Gaussian kernel is the only transform core to realize scale transformation, owns linear, shift invariant, rotation invariant etc. Mikolajczyk et. al., in the theoretical basis of the scale automatic selection, put forward with the scale invariance of the Harris operator.

**SIFT:** Scale-invariant feature transform (SIFT) is an algorithm in computer vision to detect and describe local features in images. The algorithm was patented in the US by the University of British Columbia and published by David Lowe in 1999. Applications include object recognition, robotic mapping and navigation, image stitching, 3D modeling, gesture recognition, video tracking, individual identification of wildlife and match moving.

For any object in an image, interesting points on the object can be extracted to provide a "feature description" of the object. This description, extracted from a training image, can then be used to identify the object when attempting to locate the object in a test image containing many other objects. To perform reliable recognition, it is important that the features extracted from the training image be detectable even under changes in image scale, noise and illumination. Such points usually lie on high-contrast regions of the image, such as object edges.

Another important characteristic of these features is that the relative positions between them in the original scene shouldn't change from one image to another. For example, if only the four corners of a door were used as features, they would work regardless of the door's position; but if points in the frame were also used, the recognition would fail if the door is opened or closed. Similarly, features located in articulated or flexible objects would typically not work if any change in their internal geometry happens between two images in the set being processed. However, in practice SIFT detects and uses a much larger number of features from the images, which reduces the contribution of the errors caused by these local variations in the average error of all feature matching errors.

SIFT can robustly identify objects even among clutter and under partial occlusion, because the SIFT feature descriptor is invariant to uniform scaling, orientation, and partially invariant to affine distortion and illumination changes. This section summarizes Lowe's object recognition method and mentions a few competing techniques available for object recognition under clutter and partial occlusion.

SIFT keypoints of objects are first extracted from a set of reference images and stored in a database. An object is recognized in a new image by individually comparing each feature from the new image to this database and finding candidate matching features based on Euclidean distance of their feature vectors. From the full set of matches, subsets of keypoints that agree on the object and its location, scale, and orientation in the new image are identified to filter out good matches.

The determination of consistent clusters is performed rapidly by using an efficient hash table implementation of the generalized Hough transform. Each cluster of 3 or more features that agree on an object and its pose is then subject to further detailed model verification and subsequently outliers are discarded. Finally the probability that a particular set of features indicates the presence of an object is computed, given the accuracy of fit and number of probable false matches. Object matches that pass all these tests can be identified as correct with high confidence.

**SURF:** In computer vision, Speeded Up Robust Features (SURF) is a patented local feature detector and descriptor (Bay *et al*., 2008). It can be used for tasks such as object recognition, image registration, classification or 3D reconstruction. It is partly inspired by the scale-invariant feature transform (SIFT) descriptor. The standard version of SURF is several times faster than SIFT and claimed by its authors to be more robust against different image transformations than SIFT.

To detect interest points in (Gavrila, 2000), SURF uses an integer approximation of the determinant of Hessian blob detector which can be computed with 3 integer operations using a pre-computed integral image. Its feature descriptor is based on the sum of the Haar-wavelet response around the point of interest. These can also be computed with the aid of the integral image. SURF descriptors have been used to locate and recognize objects, people or faces, to reconstruct 3D scenes, to track objects and to extract points of interest.

SURF was first presented by Herbert Bay, *et al.*, at the 2006 European Conference on Computer Vision. An application of the algorithm is patented in the United States. An "upright" version of SURF (called U-SURF) is not invariant to image rotation and therefore faster to compute and better suited for application where the camera remains more or less horizontal.

The image is transformed into coordinates, using the multi-resolution pyramid technique, to copy the original image with Pyramidal Gaussian or Laplacian Pyramid shape to obtain an image with the same size but with reduced bandwidth. This achieves a special blurring effect on the original image, called Scale-Space and ensures that the points of interest are scale invariant.

**FAST:** Features from accelerated segment test (FAST) is a corner detection method, which could be used to extract feature points and later used to track and map objects in many computer vision tasks. FAST corner detector was originally developed by Edward Rosten and Tom Drummond, and published in 2006. The most promising advantage of the FAST corner detector is its computational efficiency. Referring to its name, it is fast and indeed it is faster than many other well-known feature extraction methods, such as Difference of Gaussians (DoG) used by the SIFT, SUSAN and Harris detectors. Moreover, when machine learning techniques are applied, superior performance in terms of computation time and resources can be realized. The FAST corner detector is very suitable for real-time video processing application because of this high-speed performance.

FAST corner detector uses a circle of 16 pixels (a Bresenham circle of radius 3) to classify whether a candidate point p is actually a corner. Each pixel in the circle is labeled from integer number 1 to 16 clockwise. If a set of N contiguous pixels in the circle are all brighter than the intensity of candidate pixel p (denoted by $I_p$) plus a threshold value t or all darker than the intensity of candidate pixel p minus threshold value t, then p is classified as corner. The conditions can be written as:

- Condition 1: A set of N contiguous pixels *S*, $\forall x \in S$, the intensity of $x$ $(I_x) > I_p$ + threshold t.
- Condition 2: A set of N contiguous pixels *S*, $\forall x \in S, I_x < I_p - t$.

So when either of the two conditions is met, candidate p can be classified as a corner. There is a tradeoff of choosing N, the number of contiguous pixels and the threshold value t. On one hand the number of detected corner points should not be too many; on the other hand, the high performance should not be achieved by sacrificing computational efficiency. Without the improvement of machine learning, N is usually chosen as 12. A high-speed test method could be applied to exclude non-corner points.

**BRISK:** The key stages in Binary Robust Invariant Scalable Keypoints (BRISK) are feature detection, descriptor composition and key- point matching to the level of detail that the motivated reader can understand and reproduce in (Leutenegger *et al*., 2011). It is important to note that the modularity of the method allows the use of the BRISK detector in combination with any other keypoint descriptor and vice versa, optimizing for the desired performance and the task at hand.

### Scale-Space Keypoint Detection

With the focus on efficiency of computation, the detection methodology is inspired by the work of Mair *et al.* [10] for detecting regions of interest in the image. Their AGAST is essentially an extension for accelerated performance of the now popular FAST, proven to be a very efficient basis for feature extraction. With the aim of achieving invariance to scale which is crucial for high-quality keypoints, we go a step further by searching for maxima not only in the image plane, but also in scale-space using the FAST score s as a measure for saliency. Despite discretizing the scale axis at coarser intervals than in alternative high-performance detectors (e.g. the Fast-Hessian), the BRISK detector estimates the true scale of each keypoint in the continuous scale-space.

It is important to note here that both FAST and AGAST provide different alternatives of mask shapes for keypoint detection. In BRISK, we mostly use the 9-16 masks, which essentially require at least 9 consecutive pixels in the 16- pixel circle to either be sufficiently brighter or darker than the central pixel for the FAST criterion to be fulfilled.

Initially, the FAST 9-16 detector is applied on each octave and intra-octave separately using the same threshold T to identify potential regions of interest. Next, the points belonging to these regions are subjected to a non-maxima suppression in scale-space: firstly, the point in question needs to fulfill the maximum condition with respect to its 8 neighboring FAST scores s in the same layer. The score s is defined as the maximum threshold still considering an image point a corner. Secondly, the scores in the layer above and below will need to be lower as well. We check inside equally sized square patches: the side-length is chosen to be 2 pixels in the layer with the suspected maximum. Since the neighboring layers (and therefore its FAST scores) are represented with a different discretization, some interpolation is applied at the boundaries of the patch. depicts an example of this sampling and the maxima search

**ORB:** Oriented FAST and Rotated BRIEF builds on the well-known FAST keypoint detector and the recently-developed BRIEF descriptor. FAST features are widely used because of their computational properties. However, FAST features do not have an orientation component. In this ORB detector, efficiently computed orientation is added.

In the process of detecting interest points in the image, FAST takes one parameter, the intensity threshold between the center pixel and those in a circular ring about the center. FAST-

9 (circular radius of 9) is used in detection, which has good performance. FAST does not produce a measure of cornerness, and have found that it has large responses along edges. A Harris corner measure is employed to order the FAST keypoints. For a target number N of keypoints, first set the threshold low enough to get more than N keypoints, then order them according to the Harris measure, and pick the top N points. FAST does not produce multi-scale features. A scale pyramid of the image is used to produce FAST features (filtered by Harris) at each level in the pyramid.

BRIEF is a recent feature descriptor that uses simple binary tests between pixels in a smoothed image patch. Its performance is similar to SIFT in many respects, including robustness to lighting, blur, and perspective distortion. However, it is very sensitive to in-plane rotation.

BRIEF grew out of research that uses binary tests to train a set of classification trees. Once trained on a set of 500 or so typical keypoints, the trees can be used to return a signature for any arbitrary keypoint. In a similar manner, the tests least sensitive to orientation are observed. The classic method for finding uncorrelated tests is Principal Component Analysis; for example, it has been shown that PCA for SIFT can help remove a large amount of redundant information. However, the space of possible binary tests is too big to perform PCA and an exhaustive search is used instead.

Visual vocabulary methods use offline clustering to find exemplars that are uncorrelated and can be used in matching. These techniques might also be useful in finding uncorrelated binary tests. The closest system to ORB is multi-scale oriented patches, which proposes a multi-scale Harris keypoint and oriented patch descriptor. This descriptor is used for image stitching, and shows good rotational and scale invariance. It is not as efficient to compute as our method.

Many sampling grids are possible to compare pairs of pixel intensities. BRIEF and ORB use random pairs. BRISK uses a circular pattern where points are equally spaced on circles concentric, similar to DAISY. We propose to use the retinal sampling grid which is also circular with the difference of having higher density of points near the center. The density of points drops exponential. Each sample point needs to be smoothed to be less sensitive to noise. BRIEF and ORB use the same kernel for all points in the patch. To match the retina model, we use different kernels size for every sample points similar to BRISK.

## COMPARISION OF FEATURE DETECTORS

Performance evaluations of the selected feature detectors (Van der Wal Gooitzen *et al*., 2015) are listed in Table 1 as average processing time per frame. SIFT and its variant SURF are stable but slow. Since they are multi-scale (either subsampled or of full spatial resolutions), higher latency and more buffer allocation in hardware is less favorable. FAST is a speedy detector, but less features detected and sensitive to noise is a disadvantage. HCD is a relative speedy detector. It has stable performance on a pre-filtered or subsampled image, and is relatively easy to implement in hardware.

**Table 1: Performance of Feature Detectors**

| Detector | Description | Average Time (s) |
|---|---|---|
| SIFT | Local multi-scale level features, histogram of weighted gradient locations and orientations in blocks; Slow but robust. | 0.172 |
| FAST | Corner feature encoded by the contrast of the circle of surrounded pixels. Fast but sensitive to noise. | 0.015 |
| SURF | Approximate SIFT, integral of images and determinants of Hessian matrix are used for detection of key points. | 0.265 |
| ORB | The combination of oriented FAST and rotated BRIEF features. Fast and efficient alternative of SIFT. | 0.045 |
| HCD | Local corner detector based on thedeterminant of Harris matrix | 0.030 |

## CONCLUSION

This paper has descriptions of local invariant feature detectors and their performance evaluation based on average processing time per frame. Some of the feature detectors discussed are SIFT, SURF, FAST, BRISK, ORB and HCD. The SIFT detects many interest points but it is slow. SURF is the extension of SIFT. HCD is invariant to any change in image. FAST detector is the fastest of all detectors but susceptible to noise. ORB is combination of FAST and BRIEF descriptor.

## REFERENCES


1. Alahi A, Ortiz R and Vandergheynst P (2012), "Freak: Fast Retina Keypoint", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 510-517.
2. Bai Xinbei, Ning Xiaolin and Wang Longhua (2012), "Analysis and the Comparison of Feature Detection and Matching the Algorithms for Rovers Vision and Navigation", School of Instrument Science and the Opto–electronics Engineering, pp. 2617-2623, Beihang University Beijing, China.
3. Bay H, Ess A, Tuytelaars T and Van Gool L (2008), "Speeded Up Robust Features (SURF)", *Computer Vision and Image Understanding*, Vol. 110, No. 3, pp. 346-359.
4. Belongie S, Malik J and Puzicha J (2002), "Shape Matching and Object Recognition and Using Shape Contexts", *IEEE Transactions on Pattern of Analysis and Machine Intelligence*, Vol. 24, pp. 509-522.
5. Broggi A, Caraffi C, Fedriga R and Grisleri P (2005), "Obstacle Detection with Stereo Vision for Off- Road Vehicle Navigation", in Computer Vision and Pattern Recognition CVPR Workshop, IEEE Computer Society Conference, pp. 65-76.
6. Gavrila D M (2000), "Pedestrian Detection from a Moving Vehicle", in *Computer Vision, ECCV*, pp. 37-49, Springer.

7. Gevers T and Smeulders A W (1999), “Color Based Object Recognition”, in *Pattern Recognition*, Vol. 32, pp. 453-464.

8. Harris C and Stephens M (1998), “A Combined Comer and Edge Detector” , 4th ALVEY Vision Conference, pp. 147-151, Springer-Verlag, USA.

9. Leutenegger S, Chli M and Siegwart R Y (2011), “Brisk: Binary Robust Invariant Scalable Keypoints”, IEEE Conference on Computer Vision (ICCV), pp. 2548-2555.

10. Lowe D G (2004), “Distinctive Image Features from the Scale- Invariant Keypoints”, *International Journal of Computer Vision*, Vol.60, No. 2, pp. 91-110.

11. Rublee Ethan, Rabaud Vincent, Konolige Kurt and Bradski Gary (2011), “ORB: An Efficient Alternative to SIFT or SURF”, IEEE Conference on Computer Vision.

12. Van der Wal Gooitzen, Zhang David, Kandaswamy Indu, Marakowitz James, Kaighn Kevin, Zhang Joe and Chai Sek (2015), “FPGA Acceleration for Feature Based Processing Applications”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.

13. Viola P and Jones M (2001), “Rapid Object Detection Using a Boosted Cascade of Simple Features”, *Proc. in IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 511-518.