I J C T A, 9(11) 2016, pp. 5181-5190

A STUDY ON FEATURE DETECTORS AND DESCRIPTORS FOR OBJECT RECOGNITION

Ravi Kiran Boggavarapu [*] and Pushpendra Kumar Pateriya [**]

Abstract: Object Recognition is a fundamental problem in Vision domain. Therefore, finding unique and useful features without compromising the performance of the system is essential for any Feature Descriptor. Past decade has seen a rapid and significant improvement in Feature Descriptors. In this paper, we study and discuss seven state-of-the-art descriptors, such as SIFT, SURF, FAST, BRIEF, ORB, BRISK, and FREAK. We also briefly analyze the results of surveys that compare these descriptors. At the end of this paper, after examining the previous survey results, we conclude that binary descriptors are usually faster than their counterparts are, and BRISK outperforms all the other descriptors, followed by FREAK.

Key Words: SIFT, SURF, ORB, BRISK, FREAK, Detector, Descriptor

1. INTRODUCTION

Features are important. Many Computer Vision technologies, such as object recognition, employ feature descriptors at their core. With a huge increase in the number of low power and hand-held devices, which have the ability to capture images, the need for fast, robust, and resource-efficient feature descriptors is increasing. This advancement has augmented with tremendous research advancements for reliable and rapid feature detectors and descriptors in the past decade. In this paper, we present a brief literature study of seven feature detector and descriptor algorithms that made a huge impact in computer vision and image processing. Images Feature, or simply, Features are the pieces of information from an image, which are relevant to solve a particular computational task. Features play a vital role in Object Recognition. There can be a plethora of features in a single image, but the job of an efficient algorithm is to identify a minimum number of features that can help describe and match a particular image. These minimum numbers of features are considered as key features.

To be seen as a potential feature, it should possess some characteristics, such as - Repeatability, Distinctiveness, and robust to image transformations. Before discussing further, there is a need to identify the two types of features - Global Features and Local Features. In Global features, the contents of an image are described as a whole, thereby producing fixed-dimensional feature vectors [13]. However, global features are limited only to large-scale scenarios and fail in those scenarios, which require matching the image based on individual features in the image. Local features attempt to overcome this limitation by finding a representation that is invariant to image transformations, robust to noise, and helps identify an individual image based on its local content. Object

[*] Department of Computer Science and Engineering Lovely Professional University, Phagwara, Punjab. vsravikiran.b@gmail.com

[**] Department of Computer Science and Engineering Lovely Professional University, Phagwara, Punjab. pushpendra.14623@lpu.co.in

recognition involves Feature Detection, Feature description, and feature matching. There is a difference between feature detection and description; a feature detector identifies interest points in an image, which are typically either corners or centers of blob-like structures, and to match these points across images, we need descriptors. Descriptors are feature vectors around a particular interest point that help us identify the information from a pixel of interest.

The best place to start our discussion is, perhaps, with SIFT. Proposed by Lowe in 2004, SIFT has inspired many future detectors and descriptors and considered as the de facto go-to for many years. Even at present time, SIFT ranks high regarding performance and every new detector's performance, that was proposed in recent years, is compared with SIFT. The only limitation with SIFT is the use 128-dimensional feature vector, this reduces the performance of SIFT regarding speed and makes it less useful for real-time applications [9, 10, 11].

To overcome this limitation, SURF is introduced by Bay et al. in 2006, which attempts to address the performance issue of SIFT. SURF is fast and robust. Still, its 64-dimensional vector makes it less applicable to emerging technologies like SLAM [1, 5, 7, 9]. These performance issues in SIFT and SURF have motivated to introduce new feature descriptors that work on binary patterns; such as FAST, BRIEF, ORB, BRISK, and FREAK. Each of these algorithms attempts to address the performance issues in SIFT and SURF while designing a robust and transformation-invariant feature descriptor.

Many surveys and literature reviews have been conducted to evaluate these state-of-the-art descriptors during the past decade [9, 11, 14]. In this paper, we study these surveys and discuss the results briefly.

The paper is organized as follows: In the section that follows we study and discuss non-binary local descriptors SIFT and SURF. In section 3, we briefly discuss binary descriptors. Section 4 consists of the brief discussion on the previous survey results from [9, 11, 14]. We conclude this paper in Section 5.

2. NON-BINARY DESCRIPTORS: OVERVIEW OF SIFT AND SURF

2.1 SIFT

SIFT is a local image detector and descriptor proposed by Lowe in [8]. Unlike the previous work by Harris in [14], features extracted by SIFT are scale-invariant. The features are also invariant to rotation and contrast. The four stages of SIFT are:

1. Scale-space peak selection
2. Keypoint localization
3. Orientation assignment
4. Keypoint description

The first stage uses Difference of Gaussians (DoG) to identify the potential keypoints. Based on the work of [15], which suggests finding candidate points from the local minima and maxima in a Laplacian of Gaussian (LoG) [16]. Inspired by this idea, Lowe has approximated the Difference of Gaussian (DoG).

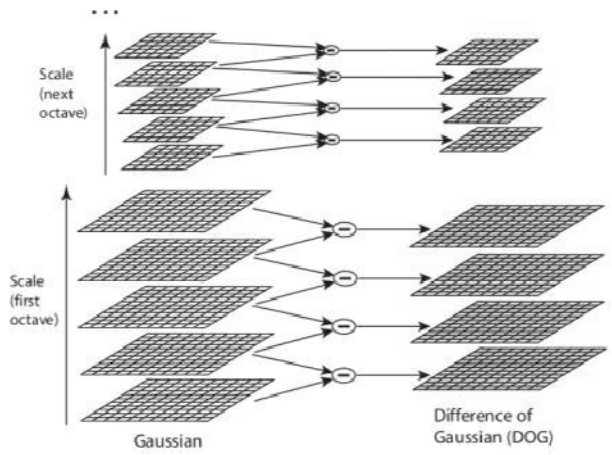

Figure 1: A Gaussian filter is repeatedly applied to the image at different scales, producing a Gaussian convoluted image, which is shown in the left-bottom corner in above figure (colored in yellow). A difference is obtained between each consecutive scale layer of convoluted image, as shown in the right-bottom corner of the above figure (colored in green). After each octave, the convoluted image is scaled down by a factor of 2 [8].

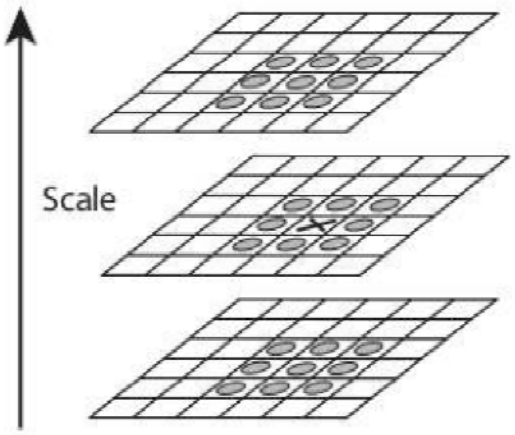

Figure 2: A pixel is selected from an octave of the convoluted image and it is compared for local extrema with the 8 surrounding pixels and also with pixels at the adjacent scales. Therefore, a pixel is compared with 26 pixels [8].

A Laplacian of Gaussian (LoG) is obtained by applying Gaussian Convolution G, to the Image I. Where s in the below equation denotes the scale. The Difference of Gaussian (DoG) is computed by taking the difference of two adjacent scales separated by a constant multiplicative factor k in each octave (as shown in figure 1).

$$L(x, y, s) = G(x, y, s) * I(x, y)$$

$$D(x, y, s) = (G(x, y, ks) - G(x, y, s)) * I(x, y)$$

$$= L(x, y, ks) - L(x, y, s)$$

In the next step, the keypoints are localized by finding local extrema, in scale and space, in the DoG images. A keypoint is considered as a stable point iff it is an extrema (as shown in figure 2), hence, eliminating the keypoints which are spatially unstable. Outliers are rejected by taking the Taylor series expansion of DoG as explained in [17]. Further, those keypoints that are unstable in scale and space are limited using the ratio of eigenvalues of the Hessian matrix. Rather than taking the eigen values (E) directly, the authors considered the ratios of Trace and Determinant of the below matrix to check principle curvature.

$$H = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{pmatrix}$$

$$E = \frac{Tr^2(H)}{Dt(H)}$$

The next step is Orientation Assignment. In this step, a principle orientation is assigned to each keypoint; while considering the direction of keypoint, the coordinates of descriptor and orientations of the gradient are rotated. In the final step, a feature vector for a keypoint descriptor is generated on regions of size 4 consisting of 8 bins each, by considering every potential feature in a 16 X 16 region (as shown in figure 3). As a process of Object recognition, the keypoints are matched individually with the database [8]. All

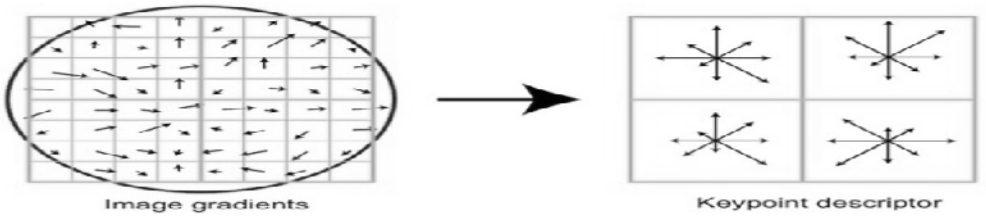

Figure 2: A keypoint descriptor is generated on a 4x4 regions consisting of 8 bins each by considering the orientation of every feature in a 16x16 region [9].

candidate keypoints are compared with the nearest neighbors and the one with the minimum Euclidian distance is considered as a potential keypoint and added to the descriptor vector [8].

2.1 SURF

SIFT, being a 128-dimensional vector, is relatively slow to compute and match. This can be a drawback for real-time applications such as SLAM [5]. SURF, introduced in [9] is robust, scale and rotation invariant, and performance wise faster than SIFT [7]. SURF approximates the LoG for Box filters and it relies on integral images as described in [18]. The use of integral images makes the computations faster. Since, the sum of the rectangles can be computed in a four array references [8]. The SURF detector is based on the Hessian matrix, and it relies on the Integral image and introduced a modification version of Hessian Matrix called "Fast-Hessian" matrix [9] is used as a Feature Detector. These authors have done indexing on the sign of Laplacian for fast matching.

$$H(x,s) = \begin{pmatrix} D_{xx}(x, s) & D_{xy}(x, s) \\ D_{yx}(x, s) & D_{yy}(x, s) \end{pmatrix}$$

Another factor contributing to the speed of detector is that SURF used box-filters on integral image. Therefore, unlike in SIFT there is no need to apply the same filter to the output of the previously filtered layer; rather, the filter can be directly applied on the original image. For orientation, the descriptor uses the information taken from a circular region around the keypoint. For a detected scale s, x and y directions are calculated on a set of pixels within a radius 6s are calculated. Finally, a square window centered at a keypoint is constructed, on which the SURF descriptor is computed and oriented. This window is further divided into sub-regions of size 4 and within each sub-region, Haar functions of size 2s are calculated [7]. Each sub-region in the window contributes four values, thus, resulting in a 64-dimensional normalized descriptor vectors.

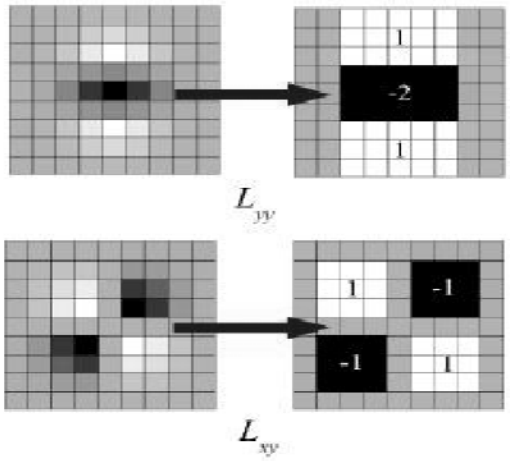

Figure 3: Each octave is smoothed using the Heat equations, applied in y- and xy-directions, and the feature values are obtained using Haar Filters [8].

3. BINARY DESCRIPTORS: OVERVIEW OF FAST, BRIEF, AND ORB

3.1 FAST

FAST (Features from Accelerated Segment Test) is a corner detection algorithm that was proposed by E. Rosten and T. Drummond in [1, 4]. It integrates machine learning and a feature detector that was described in [2, 3]. To detect features, FAST uses a segmentation test.

- 16 pixels are considered around a corner candidate '*p*', which has intensity I_p and let '*t*' be the threshold.
- *p* is considered as a corner if there exists a continuous set of n pixels, all of which are darker than I_p-t or brighter than I_p+t.
- The value of n is chosen to be 3/4th of the number of pixels considered, which in the case of 16 pixels is 12.

A high-speed test was proposed that eliminates most of the pixels, while considering only the pixels 1, 5, 9, and 13. A pixel p is considered as a corner only iff 3 of the afore-mentioned pixels are either darker than I_p-t or brighter than I_p+t.

Machine Learning Approach: As said in the previous paragraph, FAST detector uses a machine learning approach proposed as follows:

- Get the keypoints from the segmentation test.
- For every feature obtained through segmentation test, select 16 pixels around it to form feature vector P.
- Now, each pixel x in P can have one of the following three states.
- Feature vector P is divided into three subsets de-pending upon the states:

1. Darker if the intensity of p (p belongs to feature vector P) is lesser than I_p-t.
2. Similar if the intensity of p lies between I_p-t and I_p+t.
3. Brighter if the intensity of p is greater than threshold I_p+t.

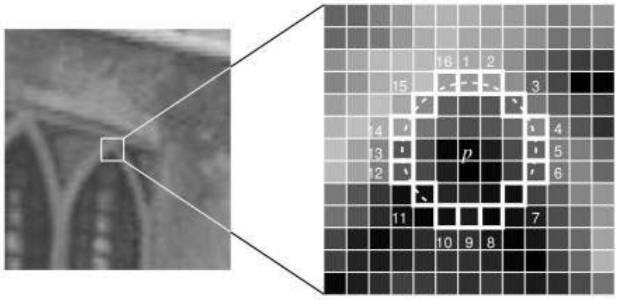

Figure 4: For a pixel candidate *p* at center, surrounding 12 contiguous pixels are shown as brighter than p.

- Finally, ID3 algorithm is used to select K_p which has knowledge about the pure classes (Pure classes are those whose Entropy E=0).
- K_p finds entropy at every iteration and selects most promising x for a particular class.

To solve the problem of adjacent keypoints, FAST uses non-maximal suppression that computes a score function V, which is the absolute difference between the keypoint and its surrounding 16 points. Between two adjacent keypoints, the one with the minimum value of V is rejected.

3.2 BRIEF

M. Calonder et al. proposed BRIEF algorithm in [5]. BRIEF uses binary strings as a feature descriptor and classifies the image patches by a relatively small number of pairwise intensity comparisons [6]. The outputs of these comparisons are stored in a bit string. A smoothed image patch p is considered, on which a binary test W is defined as:

$$W(p;x,y) = \begin{cases} 0, \text{if } p(x) < p(y) \\ 1, Otherwise \end{cases}$$

3.3 ORB

ORB proposed by E. Rublee et al. is an open-source alternative of SIFT and SURF. ORB uses the key features of FAST and BRIEF, in the sense, it uses the FAST detector, proposed in [1, 4], and BRIEF Descriptor [5]. Rublee et al. proposed and tested a modified version of FAST, which is orientation invariant. FAST fails to compute multi-scale features. The authors of ORB produced FAST features (filtered by Harris) at each level in the scale pyramid. ORB adapts the concept of intensity centroid, as proposed by Rosin in [4], to compute effectively the corner orientations, as given below. Moments of a patch are calculated as by the following formula as defined in [4]:

$$m_{pq} = \Sigma x^p y^q I(x,y)$$

$$C = \left(\frac{m10}{m11}\frac{m01}{m00}\right)$$

$$\theta = atan2(m01, m10)$$

Using the above formula, they found the centroid of the patch and calculated the orientation of the patch.

They also introduced a modified version of BRIEF algorithm [5] called rBRIEF which is a rotation-aware BRIEF - which they call steered BREIF. Note that the rBRIEF is a rotation-aware BRIEF, to which they added a learning step to find less correlated binary step.

3.4 BRISK

Proposed by S. Leutenegger et al. in [11] BRISK is a multi-scale corner detector. The major improvement in BRISK over FAST is that, the former seeks for corners not only in the original image but also in the scale space of the image. BRISK, unlike FAST, considers a number of concentric circles in a region around the interest point, as shown in figure 6. The radius of these circles decreased as going towards candidate point center. BRISK also uses binary descriptors to detect efficiently and match the keypoints and evaluates the corners based on a FAST's saliency score S. By saliency criteria, keypoints are identified throughout the image at different scale dimensions in a continuous domain. Unlike other works such as [9], BRISK does not match the descriptors based on Euclidian distance, it rather uses Hamming distance; this is one of the important factors contributed to the speed of the algorithm. Since, Hamming distance works like a charm on binary descriptors. To make the algorithm rotation-invariant, BRISK identifies the characteristic orientation of keypoint to allow for orientation-normalization. The major contribution of BRISK is that it makes use of a sampling pattern consisting of equally spaced concentric circles centered at a keypoint.

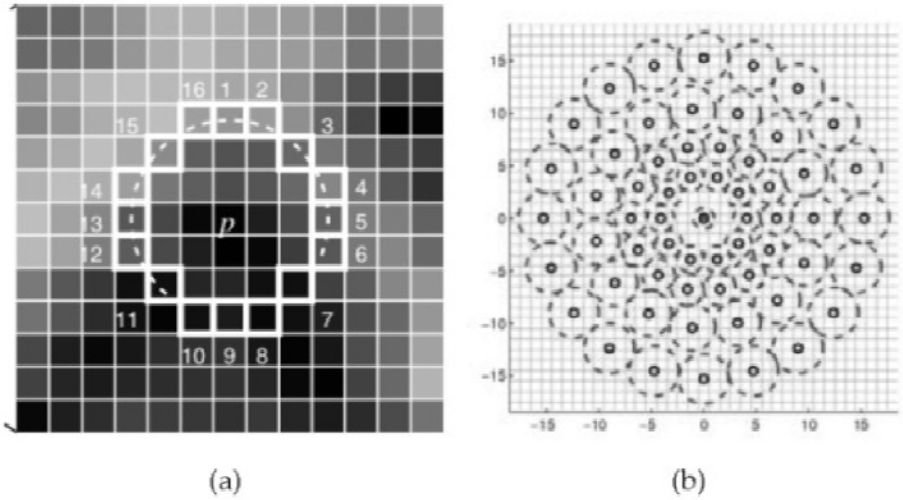

Figure 5: Brisk forms concentric circles around the pixel candidate as shown in (b), which is similar to the FAST shown in (a) [11].

3.5 FREAK

Inspired by the biological mechanism of the human eye, A. Alahi et al. proposed FREAK, dubbed as Fast Retina Keypoint, in [12]. FREAK is a binary keypoint descriptor that operates on a cascade of binary strings, which are computed by comparing image intensities over retinal sampling pattern [12]. FREAK also uses concentric circles for sampling pattern similar methodology as [11], with a difference in position of circles. The pixels being averaged (position of circles) are much more concentrated at the keypoint, i.e., center. This sampling, as claimed by authors is inspired by the retinal ganglion cells distribution with their corresponding receptive fields [12]. Retinal ganglions though vary in size, like circles in FREAK, shares a similar property. Receptive fields are represented by circles of Gaussian-convoluted image. FREAK evaluates 43 weighted Gaussians around the keypoint. Finally, local gradients are summed over selected pairs to estimate the orientation. The authors inspired by the key mechanisms of [5, 7, 12], designed a fast and robust

keypoint detector, which outperformed all the state-of-the-art keypoint detectors [7, 9, 10, 12] of the time [13].

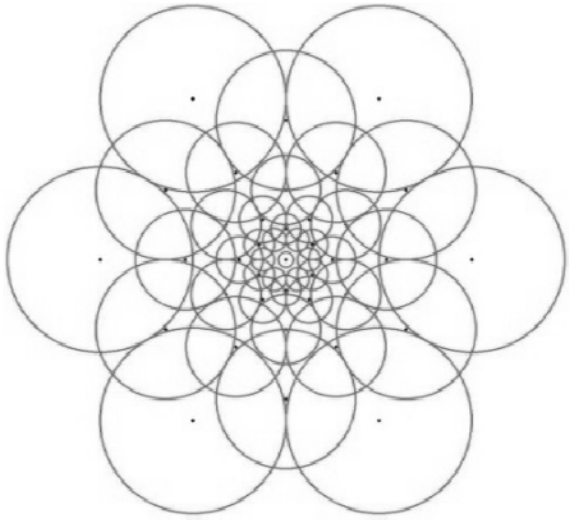

Figure 6: FREAK uses a similar strategy as BRISK (Figure 6). However, rather it forms concentric circles of different sizes, which are denser at the point of interest [12].

4. DISCUSSION ON PREVIOUS SURVEYS

As mentioned in the introduction, there are many surveys [8, 11, 14] that were conducted during the past decade to evaluate the descriptors that were mentioned in this paper. We discuss some of these important surveys. Please note that the graphs and plots presented below are directly taken from the original literature survey papers for which we give credit to the original authors. We are not claiming any results; rather we have studied and analyzed the results from these surveys.

SIFT is a good descriptor, albeit limited by its performance. SURF is considered as a fast alternative to SIFT. [7] Provided an excellent survey on SIFT and SURF comparing the two descriptors on many factors, such as invariance to rotation, detection accuracy and performance in the presence of noise. The best part of the survey in [7] is that the authors compared various dimensional variants of SIFT with SURF, such as SIFT-32, SIFT-64, SIFT-96, and the original, SIFT-128. By observing the plots in figure 8, the authors of [7] have concluded that SIFT-64 is preferred in most of the cases, especially those involving partial occlusion. SURF, although as good as SIFT, lacks during scaling, large blur and view point changes. Matching accuracy is same in all the variants of SIFT, but SIFT-64 is three-times accurate than the original SIFT-128. The authors presented the following table of that consists of matching speeds of the two descriptors. Another good survey is [10]; which submitted an evaluation of state-of-the-art binary keypoint descriptors, namely BRIEF, ORB, BRISK and FREAK. The authors obtained results, which show that BRISK outperformed all the binary descriptors, followed by FREAK, which offers comparably good result.

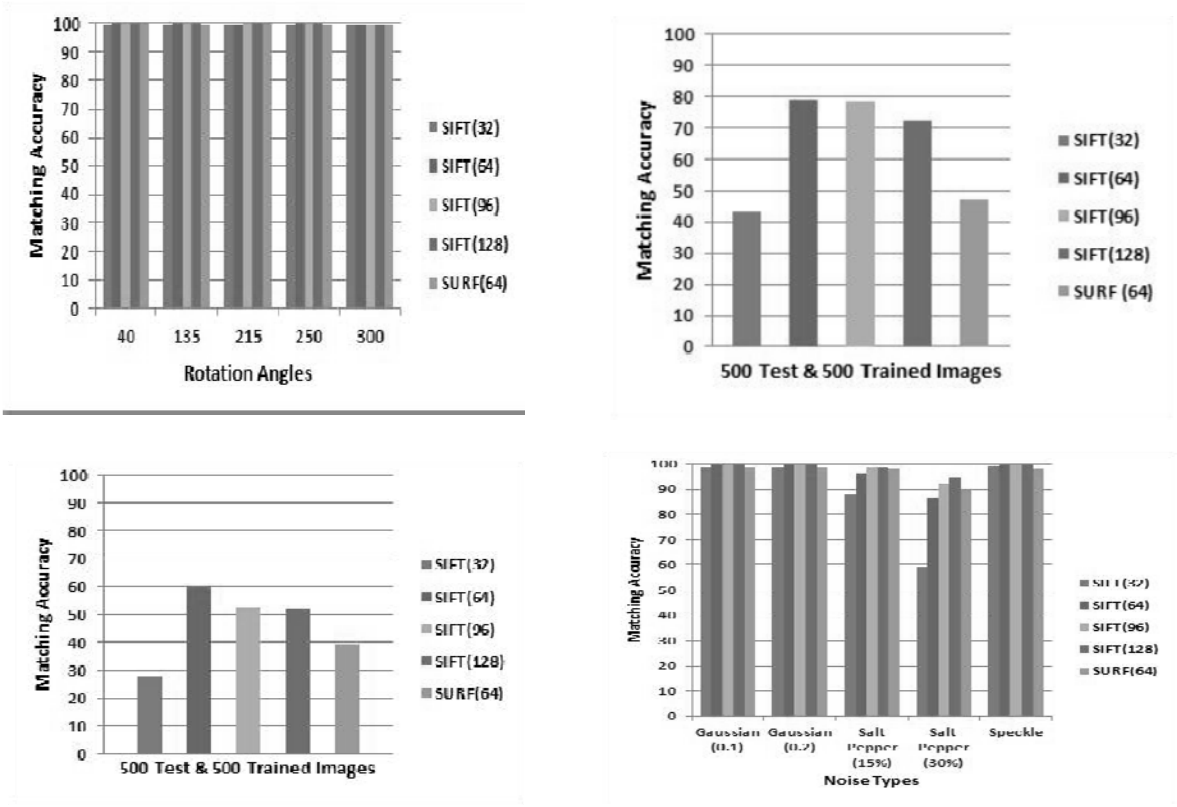

Figure 8: Figures are ordered in a right-to-left and top-to-down manner and considered as labeled from (a) to (d). (a) SIFT vs SURF performance on rotated images. (b) SIFT vs SURF performance on scaled images. (c) SIFT vs SURF performance for viewpoint changes. (d) SIFT vs SURF performance on noisy images. These results are directly taken from [7].

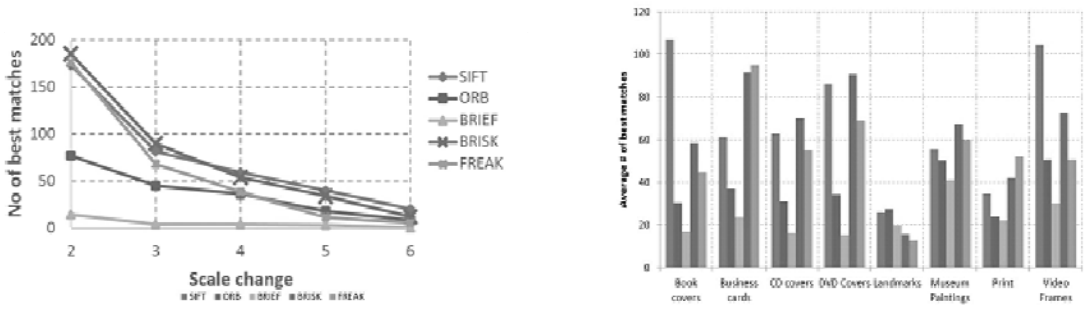

Figure 9: The plot at the top shows the performance of different descriptors in terms of number of matches for changes in scale. The bottom figure shows the average number of matches of different descriptors in different scenarios [10].

5. CONCLUSION

In this paper, we presented a literature study on various feature detectors and descriptors. We also discussed the surveys that attempt to evaluate these descriptors. After a detailed study of different descriptors and surveys comparing the descriptors, it is legible that, even though FREAK and BRISK show almost same performance, BRISK outperforms all the other state-of-the-art detectors.

References

[1] Edward Rosten and Tom Drummond Machine learning for high-speed corner detection, In Proceedings of the 9th European Conference on Computer Vision - Volume Part I (ECCV'06), 2006.

[2] Rosten, Edward; Drummond, Tom, "Fusing points and lines for high performance tracking," in Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on , vol.2, no., pp.1508-1515 Vol. 2, 17-21 Oct. 2005

[3] Edward Rosten, Gerhard Reitmayr, and Tom Drummond. 2005. Real-Time video annotations for augmented reality. In Proceedings of the First International Conference on Advances in Visual Computing (ISVC'05), George Bebis, Richard Boyle, Darko Koracin, and Bahram Parvin (Eds.). Springer-Verlag, Berlin, Heidelberg.

[4] Rosten, Edward; Porter, R.; Drummond, Tom, "Faster and Better: A Machine Learning Approach to Corner Detection," in Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.32, no.1, pp.105- 119, Jan. 2010

[5] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. 2010. BRIEF: Binary robust independent elementary features. In Proceedings of the 11th European conference on Computer vision: Part IV (ECCV'10), Kostas Daniilidis, Petros Maragos, and Nikos Paragios (Eds.). Springer-Verlag, Berlin, Heidelberg, 778-792.

[6] Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G., "ORB: An efficient alternative to SIFT or SURF," in Computer Vision (ICCV), 2011 IEEE International Conference on , vol., no., pp.2564-2571, 6-13 Nov. 2011.

[7] Khan, N.Y.; McCane, B.; Wyvill, G., "SIFT and SURF Performance Evaluation against Various Image Deformations on Benchmark Dataset," in Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on , vol., no., pp.501-506, 6-8 Dec. 2011

[8] David G. Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. Int. J. Comput. Vision 60, 2 (November 2004), 91-110.

[9] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. 2008. Speeded-Up Robust Features (SURF). Comput. Vis. Image Underst. 110, 3 (June 2008), 346-359.

[10] Bekele, D.; Teutsch, M.; Schuchert, T., "Evaluation of binary keypoint descriptors," in Image Processing (ICIP), 2013 20th IEEE International Conference on , vol., no., pp.3652-3656, 15-18 Sept. 2013

[11] Leutenegger, S.; Chli, M.; Siegwart, R.Y., "BRISK: Binary Robust invariant scalable keypoints," in Computer Vision (ICCV), 2011 IEEE International Conference on , vol., no., pp.2548-2555, 6-13 Nov. 2011.

[12] Alahi, A.; Ortiz, R.; Vandergheynst, P., "FREAK: Fast Retina Keypoint," in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on , vol., no., pp.510-517, 16-21 June 2012.

[13] Canclini, A.; Cesana, M.; Redondi, A.; Tagliasacchi, M.; Ascenso, J.; Cilla, R., "Evaluation of low-complexity visual feature detectors and descriptors," in Digital Signal Processing (DSP), 2013 18th International Conference on , vol., no., pp.1-7, 1-3 July 2013.

[14] Harris, C. Stephens, M. (1988), A Combined Corner and Edge Detector, in "Proceedings of the 4th Alvey Vision Conference", pp. 147–151.

[15] Tony Lindeberg. 1998. Feature Detection with Automatic Scale Selection. Int. J. Computer. Vision 30, 2 (November 1998), 79-116.

[16] Richard Szeliski. 2010. Computer Vision: Algorithms and Applica tions (1st ed.). Springer-Verlag New York, Inc., New York, NY, USA.

[17] Brown, M. and Lowe, D.G. 2002. Invariant features from interest point groups. In British Machine Vision Conference, Cardiff, Wales, pp. 656-665.

[18] Paul Viola and Michael J. Jones. 2004. Robust Real-Time Face Detection. Int. J. Comput. Vision 57, 2 (May 2004), 137-154.

[19] P M Panchal, S R Panchal, S K Shah. A Comparison of SIFT and SURF International Journal of Innovative Research in Computer and Communication Engineering Vol. 1, Issue 2, April 2013

[20] A Patel, D R Kasat, S Jain, V M Thakare. Performance Analysis of Various Feature Detector and Descriptor for Real-Time Video based Face Tracking. International Journal of Computer Applications (0975 8887) Volume 93 No 1, May 2014