Performance Analysis of Various Feature Detector and Descriptor for Real-Time Video based Face Tracking

Akash Patel
P.G. student
Computer Engineering
SCET, Surat, India.

D. R. Kasat
Associate Professor
SCET college
Surat, India

Sanjeev Jain, Ph. D
Director
MITS
Gwalior, India

V. M. Thakare, Ph. D
Head CSE Dept.
Amravati University
Amravati, India

ABSTRACT
This paper presents the performance analysis of various contemporary feature detector and descriptor pair for real time face tracking. These feature detectors/descriptors are mostly used in image matching applications. Some feature detectors/descriptors like STAR, FAST, BRIEF, FREAK, and ORB can also be used for SLAM applications due to their high performance. However using only one of these feature detectors for object tracking may not provide good accuracy due to various challenges in tracking like abrupt change in object motion, non-rigid object structure, change in appearance of object, occlusions in the scene and camera motion. But it can be combined other object tracking algorithm to improve the overall tracking accuracy. In this paper we have measured the tracking speed and accuracy of these feature detectors in real time video for face tracking using parameters like average number of detected key points, average detection time of key-point, frame per second and number of matches using OpenCV.

General Terms
Object tracking, Image matching.

Keywords
Face tracking, Feature detectors and Feature descriptors.

1. INTRODUCTION
Visual object tracking can be defined as the process of tracking a moving object(s) continuously using a camera. The goal of object tracking is to determine the position of the object in frames continuously and reliably in video [1]. It is very important task in many computer vision applications. This process should keep track of its motion, orientation, occlusion in scene etc. Tracking can be simplified by imposing constraints on the motion and/or appearance of objects [2].

Feature detectors are used to find interest points in given image. It aims at computing abstractions of image information whereas feature extraction aims at how to represents the detected key points of image. Feature extraction is basically a special form of dimensionality reduction. These detectors/descriptors are used as first step in many applications like object tracking, localization, image matching and recognition.

The detection, description and matching of feature points plays a vital role in most of the contemporary algorithms for SLAM (Simultaneous Localization and Mapping) [3, 4]. In past years several new detectors (FAST [6], SURF [7], and CenSurE-based STAR [8]) and descriptors (SIFT [5], SURF [7], BRIEF [9], ORB [10], BRISK [11], and FREAK [13]) have been proposed. They have been successfully applied to the object detection and tracking task.

Currently, to the extent of our knowledge there is no comparative study of the newest point detectors and descriptors with regard to their applicability in face tracking. In [14] author has compared various feature descriptors for Pedestrian detection. In [15] and [16] the authors has described the desired characteristics of the feature detectors and descriptors for visual SLAM, but they have not presented any experimental results.

This paper present the performance analysis of the detector descriptor pairs in the context of face tracking. The measure of the pair's efficiency was based on the various parameters like average number of detected key points, average detection time of key-point, detection frame per second and number of matches. The videos were taken from several real-time situations using Webcam supporting resolution up to 720p and speed up to 30 fps.

The following paper is organized as follows. The Section 2 presents the short summary of feature detector and descriptor evaluated in the study. Section 3 presents the evaluation methodology and result analysis and the section 4 contain the concluding remarks.

2. VARIOUS FEATURE DETECTORS AND DESCRIPTORS

2.1 FAST feature detector
The FAST [6] (Features from Accelerated Segment Test) feature detector was the first algorithm based on AST (Accelerated Segment Test). It first examines the values of the intensity function of pixels in a circle of radius r around the candidate point p. They have considered pixel on a circle 'bright' if its intensity value is brighter by at least t(threshold), and 'dark' if its intensity value is darker by at least t than the intensity value of p. They have classified a candidate pixel as a feature on a basis of a segment test – if a contiguous, at least n pixels long arc of 'bright' or 'dark' pixels is found in the circle than it is considered as feature. They have used ID3 [17] algorithm to optimize the order in which pixels are tested, resulting in high computational efficiency. The segment test alone produces small sets of adjacent positive responses. To further refine the results, they have used an additional corner-ness measure for non-maximum suppression (NMS). To improve the speed the NMS is applied only to a small fraction of pixels that positively passed the segment test.

2.2 SURF feature detector/descriptor

The SURF [7] (Speeded Up Robust Features) is a robust local feature detector and descriptor. It is inspired by the SIFT [5] detector/descriptor. Its main objective was to overcome SIFT's main weakness – its computational complexity and hence a low execution speed. SURF is several times faster than SIFT and it is more robust against different image transformations than SIFT as claimed by authors. The detection step in SURF takes advantage of the use of Haar wavelet approximation of the blob detector based on the Hessian determinant. The approximations of Haar wavelets can be efficiently computed using integral images, regardless of the scale. For accurate localization of multi-scale SURF features interpolation is required.

For the feature descriptor they have used Haar wavelets in conjunction with integral images to encode the distribution of pixel intensity values in the neighborhood of the detected feature while accounting of the feature's scale. They have computed the descriptor for a given feature at scale *s* which begins with the assignment of the dominant orientation to make the descriptor rotation invariant.

2.3 CenSurE based STAR feature detector

The STAR keypoint detector was implemented as a part of the OpenCV computer vision library. It is derived from CenSurE (Center Surround Extrema) feature detector [8]. The authors aimed at the formation of a multi-scale detector with full spatial resolution. As defined in [8], the subsampling performed by SIFT [5] and SURF [7] affects the accuracy of feature localization. The detector uses a bi-level approximation of the Laplacian of Gaussians (LoG) filter. The circular shape of the mask is replaced by an approximation that preserves rotational invariance and enables the use of integral images for efficient computation. They have created scale-space without interpolation, by applying masks of different size.

2.4 BRIEF corner descriptor

The BRIEF [9] (Binary Robust Independent Elementary Features) descriptor proposed in [8] uses binary strings for feature description and subsequent matching. This enables the use of Hamming distance to compute the descriptor similarity. Such similarity measure can be computed very efficiently – much faster than the commonly used L2 norm. Due to BRIEF's sensitivity to noise, the image is smoothed with a simple averaging filter before applying the actual descriptor. The value of each bit contributing to the descriptor depends on the result of a comparison between the intensity values of two points inside an image segment centered on the currently described feature. The bit corresponding to a given point pair is set to if the intensity value of the first point of this pair is higher than the intensity value of the second point, and reset otherwise. The sampling strategy for the selection of point for the pairs to be compared was selected based on experiments with uniform and Gaussian random sampling using different distribution parameters. The proposed descriptor is 512-bit long and computed over a 48 × 48 pixel image patch. The initial smoothing is performed with a 9 × 9 pixel rectangular averaging filter. The basic form of BRIEF is not invariant w.r.t. rotation.

2.5 ORB feature detector/descriptor

ORB [10] is basically a fusion of FAST (Features from Accelerated Segment Test) [6] keypoint detector and BRIEF (Binary Robust Independent Elementary Features) [9] descriptor with many modifications to enhance the performance. It uses FAST to find keypoints, and then apply Harris corner measure to find top N points among them. It also use pyramid to produce multiscale-features. But one problem is that, FAST doesn't compute the orientation. So, Authors came up with following modification. It computes the intensity weighted centroid of the patch with located corner at center. The direction of the vector from this corner point to centroid gives the orientation. To improve the rotation invariance, moments are computed with x and y which should be in a circular region of radius r, where r is the size of the patch.

For descriptor, ORB uses modified version of BRIEF descriptor. Standard BRIEF descriptor performs poorly with rotation. So ORB "steer" BRIEF according to the orientation of keypoints. For any feature set of n binary tests at location (x_i, y_i), define a $2 \times n$ matrix, S which contains the coordinates of these pixels. Then using the orientation of patch, θ, its rotation matrix is found and rotates the S to get steered(rotated) version S_θ. ORB discretize the angle to increments of $2\pi/30$ (12 degrees), and construct a lookup table of pre-computed BRIEF patterns. As long as the keypoint orientation θ is consistent across views, the correct set of points S_θ will be used to compute its descriptor.

2.6 BRISK feature detector/descriptor

The BRISK [11] is a keypoint detector and descriptor inspired by AGAST [12] and BRIEF [9]. For detecting the features it uses AGAST [12] which is improvement of FAST in speed while maintaining the same detection performance. To achieve scale invariance, it detects the keypoints in a scale-space pyramid, performing non-maxima suppression and interpolation across all scales. Instead of using learned or random pattern like in BRIEF and ORB they have used symmetric pattern to describe the features. They have used several long-distance sample point comparisons to determine orientation and for long-distance comparison the vector displacement between the sample points is stored and weighted by the relative difference in intensity. Then, to determine the dominant gradient direction of patch these weighted vectors are averaged.

2.7 FREAK feature descriptor

The FREAK [13] (Fast Retina Keypoint) is a novel descriptor biologically inspired by human visual system. It provides the descriptor with feature orientation by summing the estimated local gradients over selected point pairs. It uses a specific point sampling pattern that allows applying coarser discretization of rotation, which allows savings in memory space. They have used a special, biologically inspired sampling pattern. While the resulting descriptor is still a binary string like BRIEF [9], the sampling pattern allows for the use of a 'coarse-to-fine' approach to feature description. It first compares the point pairs carrying the information on most distinctive characteristics of the feature neighborhood. This allows for faster rejection of false matches and shortening of the computation time.

3. EXPERIMENTS

3.1 Dataset

We have used our own dataset for testing various detector-descriptor pairs. We have tested each pair in several real-world situations. The face was moved left/right to test the effect of rotation for each detector/descriptor pair. The Logitech C270 Webcam supporting resolution up to 720p and speed up to 30 fps is used for taking the videos.

3.2 Evaluation

The OpenCV C/C++ library for Windows is used to perform all the tests. All the tests were executed on a laptop with an Intel 2nd gen core-i5 2430M 2.4GHz processor and 4GB RAM. The video was captured at 640 × 480 resolution.

The following procedure is adopted for testing each detector-descriptor pair:

1. The user selects the interest object from the live video.
2. The selected area is cropped from frame and it is considered as object image.
3. The selected point feature detector is applied on object image and current video frame (i.e. scene).
4. The point features descriptors are calculated for both images using the selected descriptor algorithm.
5. The features from both images are matched using hamming distance based brute-force matcher function by minimizing the distance between their descriptors.
6. The distance d between descriptor of object image and scene image is calculated.
7. The mean of this distance array is calculated using this formula:

$$mean(\mu) = \frac{\sum_{i=1}^{N} d_i}{N} \quad (1)$$

   Where N is total number of descriptors

8. The deviation of distance array is calculated using this formula:

$$deviation(\sigma) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i - \mu)^2} \quad (2)$$

9. Then the parameters like average number of detected key points, Number of matches and Time taken per frame are also calculated for each pair.

Here the mean and deviation of distance array are used as efficiency measure. The larger mean shows that the average distance between descriptor of reference image and scene image is large. So it points towards lesser efficiency of matching. While the deviation represents the average amount of difference between other descriptor value and the mean value. The deviation tends to increase when object is moved in the scene.

Firstly the test was performed on the facial images having plain background in good lighting. At first we have measured all the parameters as described above for each detector-descriptor pair on straight face. Then the face is moved left and right as shown in fig. 3.1 and again all the parameters are measured. The same test is repeated for 3 times for each pair. The results of this test are shown in table 3.1-3.5. Then we have performed similar tests in low light condition. The results of low-light test are shown in table 3.6-3.10. In following tables the number of detected keypoints and number of matches shows the average value. For mean and deviation the range is shown. The fps field shows the average frame per second of video. The distance mean and deviation value increases as the face moves left/right. The FREAK descriptor has highest deviation while the SURF detector/descriptor has lowest average mean distance and deviation but its performance is slower than all and it detects less number of keypoints than others. Binary vector descriptor BRIEF and ORB are showing good performance. In low light condition the number of detected keypoints decreases drastically for FAST and ORB detectors. BRISK detector was tested on low threshold (T=10) for low light condition because at default threshold (T=30) it was not detecting any keypoint in the face.

Table 3.1 Test results for FAST detector

Descriptor	Number of detected keypoints	Number of matches	Mean	Deviation	Fps
BRIEF	361	246	18-33	16-25	30
BRISK	350	273	71-99	39-46	30
ORB	365	193	16-31	19-27	30
FREAK	323	204	41-66	38-50	30

Table 3.2 Test results for SURF detector

Descriptor	Number of detected keypoints	Number of matches	Mean	Deviation	Fps
BRIEF	62	52	3-10	9-21	15
BRISK	58	36	7-15	26-41	15
ORB	60	52	4-10	12-21	15
FREAK	60	15	5-7	20-28	15

Table 3.3 Test results for STAR detector

Descriptor	Number of detected keypoints	Number of matches	Mean	Deviation	Fps
BRIEF	31	31	2-6	8-19	30
BRISK	32	32	7-13	24-40	30
ORB	32	32	2-6	9-18	30
FREAK	31	29	7-9	21-29	30

Table 3.4 Test results for ORB detector

Descriptor	Number of detected keypoints	Number of matches	Mean	Deviation	Fps
ORB	118	118	16-29	17-30	30
BRISK	150	95	15-42	40-63	30
FREAK	140	11	2-5	11-22	30

Table 3.5 Test results for BRISK detector

Descriptor	Number of detected keypoints	Number of matches	Mean	Deviation	Fps
BRISK	18	18	5-10	24-42	30
BRIEF	18	18	1-3	7-15	30
FREAK	17	10	2-5	16-25	30

Table 3.6 Test results for FAST detector (low-light)

Descriptor	Number of detected keypoints	Number of matches	Mean	Deviation	Fps
BRIEF	37	34	3-10	11-19	30
BRISK	48	46	13-25	36-62	30
ORB	59	55	6-14	16-32	30
FREAK	51	50	15-18	36-44	30

Table 3.7 Test results for SURF detector (low-light)

Descriptor	Number of detected keypoints	Number of matches	Mean	Deviation	Fps
BRIEF	23	23	3-5	12-17	30
BRISK	25	25	6-9	23-33	30
ORB	29	29	3-5	12-17	30
FREAK	32	32	4-6	14-18	30

Table 3.8 Test results for STAR detector (low-light)

Descriptor	Number of detected keypoints	Number of matches	Mean	Deviation	Fps
BRIEF	23	23	3-5	12-17	30
BRISK	25	25	6-9	23-33	30
ORB	29	29	3-5	12-17	30
FREAK	32	32	4-6	14-18	30

Table 3.9 Test results for ORB detector (low-light)

Descriptor	Number of detected keypoints	Number of matches	Mean	Deviation	Fps
ORB	20	20	3-4	15-18	30
BRISK	17	11	2-6	14-36	30
FREAK	18	2	2-5	10-24	30

Table 3.10 Test results for BRISK detector (low-light)

Descriptor	Number of detected keypoints	Number of matches	Mean	Deviation	Fps
BRISK	18	18	8-9	34-41	30
BRIEF	15	15	1-2	10-13	30
FREAK	15	13	2-4	11-17	30

The Fig. 3-1 shows the tracking result for FAST/BRIEF pair in good light condition. While the Fig. 3-2 shows the tracking result in low light condition. From both the fig. we can say that the number matches are very less in low light condition.

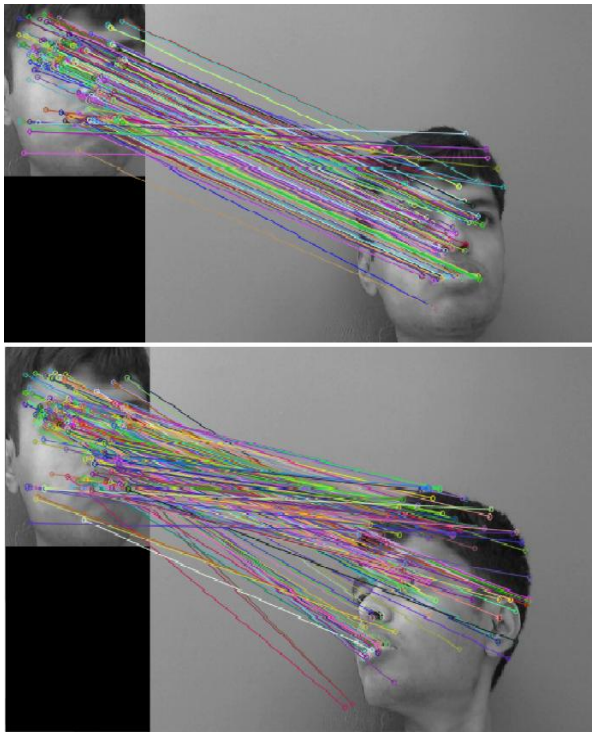

Figure 3-1 FAST/BRIEF (In good lighting)

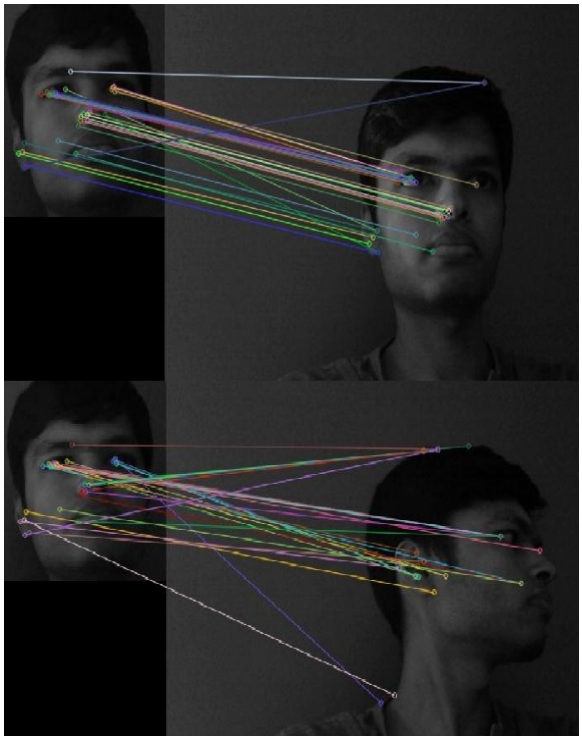

Figure 3-2 FAST/BRIEF (In low light)

4. CONCLUSIONS

We have compared various contemporary feature detector and descriptor pair to find the best combination for real time visual face tracking. The experiments show that in low light condition number of detected keypoints and matches are decreasing. The binary descriptors BRIEF and ORB are showing good performance with detectors like FAST and STAR. While the recently proposed FREAK removes so many detected keypoint when combined with SURF, ORB and BRISK and it has more deviation compared to other when object moves so it is not showing consistent performance as claimed in [13]. The SURF detector has the lowest distance deviation and mean so it is accurate. But it takes almost double time than other detectors. So it is less suitable for SLAM applications. So in short FAST/BRIEF or ORB is more suitable for real time visual face tracking.

5. REFERENCES

[1] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors", IEEE Trans. Syst. Man Cyber.-C vol. 34 (3), 2004, pp. 334–352.

[2] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. ACM Computing Survey, vol. 38(4), 2006.

[3] J. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM", IEEE Trans. PAMI, vol. 29(6), 2007, pp. 1052-1067.

[4] A. Schmidt, A. Kasiński, "The Visual SLAM System for a Hexapod Robot", Lecture Notes in Computer Science, vol. 6375, 2010, pp. 260–267.

[5] D. Lowe, "Object recognition from local scale-invariant features", in: Proceedings of the International Conference on Computer Vision ICCV, Corfu, 1999, pp. 1150–1157.

[6] E. Rosten, and T. Drummond, "Machine learning for highspeed corner detection", in Proc. of European Conf. on Computer Vision, 2006, pp. 430–443.

[7] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding, vol. 110(3), 2008, pp. 346–359.

[8] M. Agrawal, K. Konolige, and M.R. Blas, "CenSurE: Center surround extremas for real time feature detection and matching", Lecture Notes in Computer Science, vol. 5305, 2008, pp. 102–115.

[9] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary Robust Independent Elementary Features", in Proceedings of ECCV 2010, pp. 778–792.

[10] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "ORB: An efficient alternative to SIFT or SURF", in Proc. ICCV, 2011, pp. 2564–2571.

[11] S. Leutenegger, M. Chli, and R. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in Proc. Int. Conf. Computer Vision, 2011, pp. 2548–2555.

[12] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test", In Proceedings of the European Conference on Computer Vision (ECCV), 2010.

[13] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast Retina Keypoint", In Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 510-517.

[14] Schaeffer, Cameron. "A Comparison of Keypoint Descriptors in the Context of Pedestrian Detection: FREAK vs. SURF vs. BRISK", 2013.

[15] O. Martínez, A. Gil, M. Ballesta, and O. Reinoso, "Interest Point Detectors for Visual SLAM", In Current Topics in Artificial Intelligence, Springer Berlin Heidelberg, 2007, pp. 170-179.

[16] M. Ballesta, A. Gil, O. Martínez, and O. Reinoso, "Local Descriptors for Visual SLAM", in Proc. Workshop on Robotics and Mathematics, 2007.

[17] Quinlan, J.R., "Induction of decision tree", Machine learning, vol. 1(1) 1986, pp. 81-106.