



# Comparative Study of Corner and Feature Extractors for Real-Time Object Recognition in Image Processing

Arpita Mohapatra<sup>1</sup>, Sunita Sarangi<sup>1</sup>, Srikanta Patnaik<sup>2</sup>, and Sukant Sabut<sup>1\*</sup>, *Member, KIICE*

<sup>1</sup>Dept. of Electronics & Instrumentation Engineering, Institute of Technology Education & Research, SOA University, India

<sup>2</sup>Dept. of Computer Science & Engineering, Institute of Technology Education & Research, SOA University, India

## Abstract

Corner detection and feature extraction are essential aspects of computer vision problems such as object recognition and tracking. Feature detectors such as Scale Invariant Feature Transform (SIFT) yields high quality features but computationally intensive for use in real-time applications. The Features from Accelerated Segment Test (FAST) detector provides faster feature computation by extracting only corner information in recognising an object. In this paper we have analyzed the efficient object detection algorithms with respect to efficiency, quality and robustness by comparing characteristics of image detectors for corner detector and feature extractors. The simulated result shows that compared to conventional SIFT algorithm, the object recognition system based on the FAST corner detector yields increased speed and low performance degradation. The average time to find keypoints in SIFT method is about 0.116 seconds for extracting 2169 keypoints. Similarly the average time to find corner points was 0.651 seconds for detecting 1714 keypoints in FAST methods at threshold 30. Thus the FAST method detects corner points faster with better quality images for object recognition.

**Index Terms:** Computer vision, FAST, Feature extraction, Object recognition, SIFT

## I. INTRODUCTION

Object detection and recognition is becoming one of the major research areas in the field of computer vision. Feature points detection in images has become essential for several tasks in machine vision. Wide real-time applications can be found particularly in human-computer interaction, visual surveillance, robot navigation, and many more such fields. Further, feature points provide a sufficient constraint to compute image displacements reliably, and by processing these, the data are reduced by several orders of magnitude as compared to the original image data. These features could be shape, texture, colour, edge, corner, and blob.

The scale invariant feature transform (SIFT) algorithm

extracts features that are invariant to image scaling and rotation, and partially invariant to changes in illumination and three-dimensional (3D) camera viewpoints [1]. They are well localized in both spatial and frequency domains, reducing the probability of disruption by occlusion, clutter, or noise. SIFT allows the correct matching of a single feature, with high probability, against a large database, which makes this algorithm highly distinctive, for providing a basis for object and scene recognition. Due to the emerging applications of keypoint matching-based object detection and the development of SIFT, a number of detection algorithms are focusing on invariant feature-based object detection. Some of the efficient object detection algorithms are Speeded Up Feature Transform (SuRF), Oriented Binary

Received 07 April 2014, Revised 12 May 2014, Accepted 16 June 2014

\*Corresponding Author: Sukanta Sabut (E-mail: [sukantsabut@soauniversity.ac.in](mailto:sukantsabut@soauniversity.ac.in), Tel: +91-8763070027)

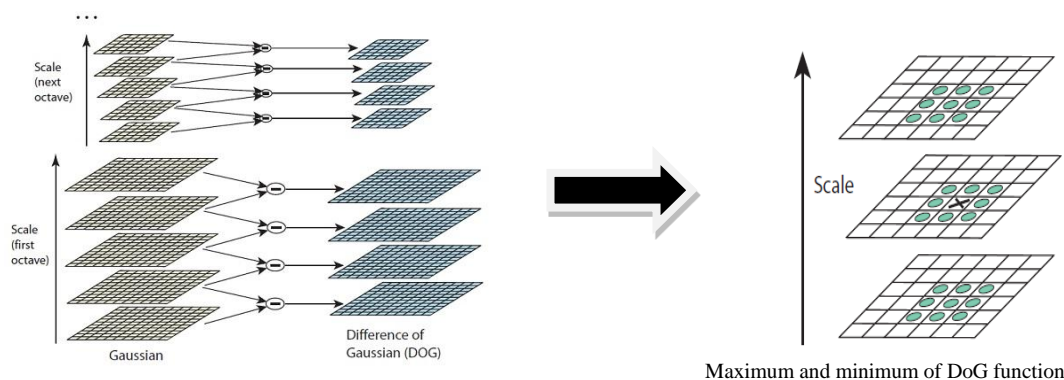
Dept. of Electronics & Instrumentation Engineering, Institute of Technology Education & Research, SOA University, India.

**Open Access** <http://dx.doi.org/10.6109/jicce.2014.12.4.263>

print ISSN: 2234-8255 online ISSN: 2234-8883

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright © The Korea Institute of Information and Communication Engineering



**Fig. 1.** Process of extracting DoG values.

Robust Independent Elementary Features (ORB), and features from accelerated segment test (FAST) [2-4].

One of the most intuitive types of feature points is the *corner* that shows a strong two-dimensional intensity change, and is well distinguished from the neighbouring points. Corner detection is used as the first step of many vision tasks, such as tracking, localization, simultaneous localisation and mapping (SLAM), image matching and recognition. Corner detectors have been widely used as feature point detectors because corners correspond to image locations with high information content and can be matched between images reliably [5]. These matched feature point locations are then taken as an input to high-level computer vision tasks. FAST is a machine learning process for the identification of interest points in an image [6]. An interest point in an image is a pixel with a defined position and has local information that is repeatable between different images. The corner response function (CRF) can be interpreted to be invariant in scale, rotation, or even affine transformations for some applications. In this study, we analyse and compare the performance of the SIFT on the basis of feature extraction and the FAST corner detection algorithm for object recognition in terms of its efficiency, quality, and robustness.

## II. METHODOLOGY

Object detection techniques detect the main points of an object in an image by dividing the image into three stages. The first stage examines the representation of features required for object recognition based on local or global image information. The second stage classifies the image based on the extracted features. The last stage is the recognition of the new image based on machine learning, which is performed by training images. The first step of object recognition is feature extraction that is used to detect the interest point of the image. The SIFT method is used to detect invariant feature points of an image.

### A. Feature Detection Using SIFT

Lowe has demonstrated that SIFT features are invariant to image scaling, rotation, and translation, and partially invariant to changes in illumination and 3D viewpoints [1]. The following are the major stages of computation used to generate the set of image features:

#### 1) Scale-Space Extrema Detection

The first stage of keypoint detection is to identify locations and scales that can be repeatedly assigned under differing views of the same object. Detecting locations that are invariant to the scale change of the image can be accomplished by searching for stable features across all possible scales, using a continuous function of scale known as scale space [7]. The advantages of using this function are its computation efficiency and close approximation to the scale-normalized Laplacian of Gaussian,  $\sigma^2 \nabla^2 G$  [8]. For efficient detection of stable keypoint locations in the scale space, it has been proposed using the scale-space extrema in the DoG function convolved with the image, computed from the difference of two nearby scales separated by a constant multiplicative factor of 'k'.

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma), \tag{1}$$

where

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \text{ (Gaussian function) } \tag{2}$$

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \text{ (scale-space function) } \tag{3}$$

and  $I(x, y)$  is the image function.

#### 2) Accurate Keypoint Localization

Once a keypoint candidate has been found by comparing a pixel to its neighbours (Fig. 1), a detailed model is fit to the nearby data for location, scale, and ratio of principal curvatures. According to the information obtained selected

points that have low contrast (and are therefore sensitive to noise) or are poorly localized along an edge are rejected. Keypoints are selected on the basis of measures of their stability. A 3D quadratic function can be fitted to the local sample points to determine the interpolated location of the maximum that provides a substantial improvement to matching and stability [9]. For determining the location of the extremum, Taylor's expansion of the scale-space function is utilized to derive an offset that sets the threshold of the extremum.

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial^2 x} x, \quad (4)$$

where  $D$  and its derivatives are evaluated at the sample point and  $x = (x, y, \sigma)^T$  is the offset from this point. The location of the extremum,  $\hat{x}$ , is given by

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x}. \quad (5)$$

The function value at the extremum,  $D(\hat{x})$ , is useful for rejecting the unstable extrema with low contrast. This is shown as

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x}. \quad (6)$$

For stability, it is not sufficient to reject keypoints with low contrast. The difference-of-Gaussian (DoG) function will have a strong response along edges, even if the location along the edge is poorly determined and therefore unstable to small amounts of noise. To eliminate the principal curvature that is found to be lying mostly at the edges, a Hessian matrix is used for evaluation. The process of eliminating the edge response includes computing the Hessian and its trace and then, checking it with a threshold set by the ratio of the trace of Hessian and the product of its determinant, given by

$$\begin{aligned} Tr(H) &= D_{xx} + D_{yy} = \alpha + \beta, \\ Det(H) &= D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta, \end{aligned} \quad (7)$$

and the ratio given by

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r}, \quad (8)$$

where  $r$  is the ratio between the largest magnitude eigenvalue ( $\alpha$ ) and the smaller one ( $\beta$ ), so that  $\alpha = r\beta$ .

### 3) Orientation Assignment

Based on local image gradient directions, one or more orientations are assigned to each keypoint location. Due to

this, the keypoint descriptor can be represented with respect to this orientation and therefore, achieve invariance to image rotation [1, 10]. The scale of the keypoint is used to select the Gaussian smoothed image,  $L$ , with the closest scale, so that all computations are performed in a scale-invariant manner. For each image sample,  $L(x, y)$ , at this scale, the gradient magnitude,  $m(x, y)$ , and orientation,  $\theta(x, y)$ , are precomputed using pixel differences:

$$\begin{aligned} m(x, y) &= \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \\ \theta(x, y) &= \tan^{-1}((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y))). \end{aligned} \quad (9)$$

The gradient orientations of sample points that lie within a region around the keypoint form the orientation histogram. Prior to the addition to the histogram, each sample is weighted by its gradient magnitude and by a Gaussian-weighted circular window when  $\sigma$  is 1.5 times that of the scale of the keypoint.

### 4) Keypoint Descriptor

The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and changes in illumination.

A descriptor is computed for the local image region that is highly distinctive yet is as invariant as possible to the remaining variations, such as changes in illumination or 3D viewpoints. A complex cell model approach based on a model of biological vision [11] provided better matching and recognition than a traditional correlation of image gradients. A keypoint descriptor is created by computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location (Fig. 2). These are weighted by a Gaussian window with  $\sigma$  equal to one half of the width of the descriptor window, shown as a circle. The orientation histograms accumulate these samples, summarizing the contents over  $4 \times 4$  sub-regions, with the length of each arrow corresponding to the sum of the gradient magnitudes in that direction within the region.

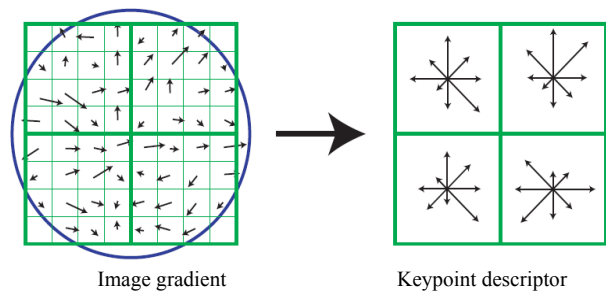


Fig. 2. Image showing a  $2 \times 2$  descriptor array computed from an  $8 \times 8$  set of samples.

Finally, the feature vector is modified to reduce the effects of illumination change. The descriptor is invariant to affine changes in illumination.

In the application to object recognition in presence of occlusion and clutter, clusters of at least three features are required. This procedure involves three stages:

- (1) First, SIFT features are extracted from a set of training images and stored in a database. A fast nearest neighbour algorithm is applied to match the SIFT features from the new or transformed image to the previous database. Features that do not hold a good match due to background clutter, are successfully discarded, which makes the computation even faster. For efficient nearest neighbour identification, a priority search algorithm called the best-bin first algorithm is used [12].
- (2) Second, clusters of at least three features are identified that agree on an object and its transformation. For this, Hough transform is used for a better performance [13, 14].
- (3) Each cluster created by Hough transform is then subject to a geometric verification using the least squares solution for consistent motion parameters.

### B. Feature Detection Using FAST

FAST is used for high-speed corner detection. The segment test criterion operates by considering a circle of sixteen pixels around the corner candidate 'p', as shown in Fig. 3.

The detailed algorithm is explained below:

1. Select a pixel 'p' in the image with an intensity of  $I_p$ . This is the pixel which is to be identified as an interest point or not. (Refer to Fig. 3)
2. Set a threshold intensity value T, (say 20% of the pixel under test).
3. Consider a circle of 16 pixels surrounding the pixel p (a Bresenham circle of radius 3).
4. To determine p as a corner, there must exist a set of 'n' contiguous pixels in the circle, which are brighter than  $I_p + T$ , or all darker than  $I_p - T$  (n = 12 admits a high-speed test which can be used to exclude a very large number of non-corners).
5. The high-speed test examines pixels (1, 9) and (5, 13) to be either brighter or darker than  $I_p$  by a factor T. For p to be a corner, at least three of these must satisfy the test; else, p cannot be a corner.
6. If at least three of the pixels are above or below  $I_p + T$ , then check for all 16 pixels in the circle and check whether 12 contiguous pixels fall in the criterion.
7. Full segment test criterion is then checked by examining all the pixels in the image.

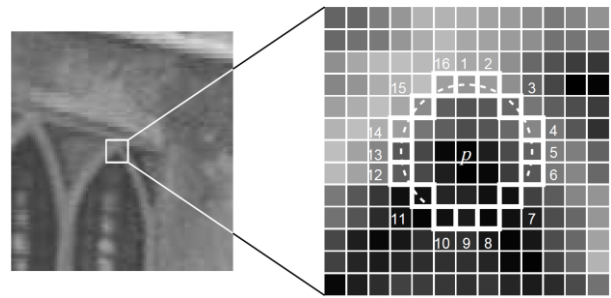


Fig. 3. Image showing the interest point under test and the 16 pixels on the circle (image copied from [5]).

Although the detector exhibits high performance in itself, there are a few limitations. First, this high-speed test does not reject as many candidates for  $n < 12$ , since the point can be a corner if only two out of the four pixels are significantly brighter and darker than p. Second, querying and distribution of the corner appearances determine the efficiency of the detector. Third, multiple feature points are detected adjacent to one another. Thus, a machine learning approach is introduced for improving the generality and speed of the detector [6, 15].

#### 1) Improving Generality and Speed With Machine Learning

The process operates in two stages. First, a set of images is taken for training. To build a corner detector for a given n, all of the 16 pixel rings are extracted from the set of images. In every image, run the FAST algorithm to detect the interest points by taking one pixel at a time and evaluating all the 16 pixels in the circle. For every pixel 'p', store the 16 pixels surrounding it as a vector, as shown in Fig. 4. The vector P contains all the data for training. This procedure is repeated for all the pixels in all the images.

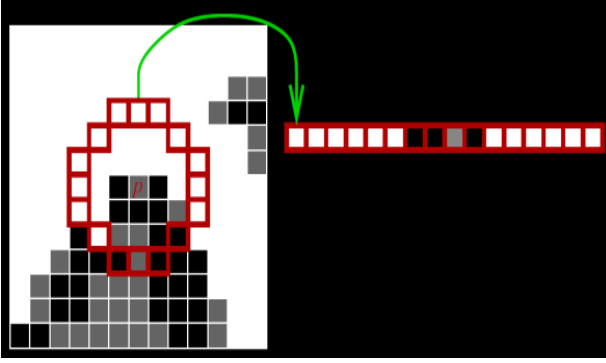
For each location on the circle  $x \in \{1 \dots 16\}$ , the pixel at this position with respect to p, denoted by  $p \rightarrow x$ , can have one of three states:

$$S_{p \rightarrow x} = \begin{cases} d, & I_{p \rightarrow x} \leq I_p - T & \text{(darker)} \\ s, & I_p - T < I_{p \rightarrow x} < I_p + T & \text{(similar)} \\ b, & I_p + T \leq I_{p \rightarrow x} & \text{(brighter)}. \end{cases} \quad (10)$$

Let P be the set of all pixels in all training images. Choosing an x partitions P into three subsets,  $P_d$ ,  $P_s$ , and  $P_b$ , where

$$P_b = \{p \in P : S_{p \rightarrow x} = b\}, \quad (11)$$

and  $P_d$  and  $P_s$  are defined similarly.



**Fig. 4.** Vector showing the 16 values surrounding the pixel  $p$  (\*image copied from [6]).

Let  $K_p$  be a Boolean variable which is true if  $p$  is a corner and false otherwise. Stage 2 employs the algorithm used in ID3 (decision tree classifier) [16, 17]. This algorithm works on the principle of entropy minimization that begins by selecting the  $x$  which contains maximum information about whether the candidate pixel is a corner, as measured by the entropy of  $K_p$ .

The total entropy of  $K$  for an arbitrary set of corners,  $Q$ , is

$$H(Q) = (c + \bar{c}) \log 2(c + \bar{c}) - c \log 2c - \bar{c} \log 2\bar{c}, \quad (12)$$

where

$$c = \{i \in Q : Ki \text{ is true}\} \quad (\text{number of corners})$$

$$\bar{c} = \{i \in Q : Ki \text{ is false}\} \quad (\text{number of non-corners})$$

The choice of  $x$  then yields the information gain ( $H_g$ ):

$$H_g = H(P) - H(P_d) - H(P_s) - H(P_b). \quad (13)$$

After selecting the  $x$  which yields the most information, the process of entropy minimization is applied recursively on all three subsets, i.e.  $x_b$  is selected to partition  $P_b$  into  $P_{b,d}$ ,  $P_{b,s}$ , and  $P_{b,b}$ ;  $x_s$  is selected to partition  $P_s$  into  $P_{s,d}$ ,  $P_{s,s}$ , and  $P_{s,b}$ ; and so on, where each  $x$  is chosen to yield maximum information about the set to which it is applied. The process is terminated when the entropy of a subset becomes zero. For a faster detection in other images, the order of querying which is learned by the decision tree can be used.

## 2) Non-maximal Suppression

Since the data contain incomplete coverage of all possible corners, the learned detector cannot detect multiple interest points. To ensure detection exactly similar to the segment test criterion in the case of FAST- $n$  detectors, an instance of every possible combination of pixels with low weight is

included. Non-maximal suppression cannot be applied directly to the resulting features as the segment test does not compute a corner response function. For a given  $n$ , as the threshold is increased, the number of detected corners will decrease. The corner strength  $T$  is defined as the maximum value for which a point is detected as a corner. The decision tree classifier can efficiently determine the class of a pixel for a given value of  $T$ . The class of a pixel (for example, 1 for a corner, 0 for a non-corner) is a monotonically decreasing function of  $T$ . Therefore, we can use bisection to efficiently find the point where the function changes from 1 to 0. This point gives us the largest value of  $T$  for which the point is detected as a corner. Since  $T$  is discrete, this is the binary search algorithm.

Alternatively, an iteration scheme can be used. A score function  $V$  is computed for each of the detected points (using midpoint circle algorithm). The score function can be defined as ‘the sum of the absolute difference between the pixels in the contiguous arc and the centre pixel’. The  $V$  values of two adjacent interest points are compared, and the one with the lower  $V$  value is discarded. Mathematically, this can be represented as

$$V = \max \begin{cases} \sum (\text{pixel values} - p) \text{ if } (\text{value} - p) > T \\ \sum (p - \text{pixel values}) \text{ if } (p - \text{value}) > T \end{cases} \quad (14)$$

where  $p$  is the central pixel,  $T$  is the threshold for detection, and pixel values correspond to the  $n$  contiguous pixels in the circle.

The score function can also be defined in alternative ways. The keypoint here is to define a heuristic function which can compare two adjacent detected corners and eliminate the comparatively insignificant one. The speed depends strongly on the learned tree and the specific processor architecture. Non-maximal suppression is performed in a  $3 \times 3$  mask.

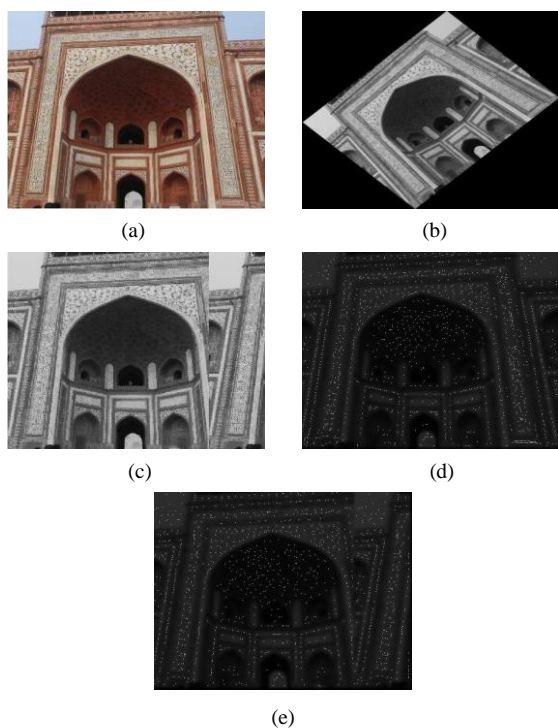
## III. EXPERIMENTAL RESULTS AND DISCUSSION

Experiments using SIFT and FAST algorithms were carried out by taking a sample image in greyscale using the MATLAB R13b computer vision tool.

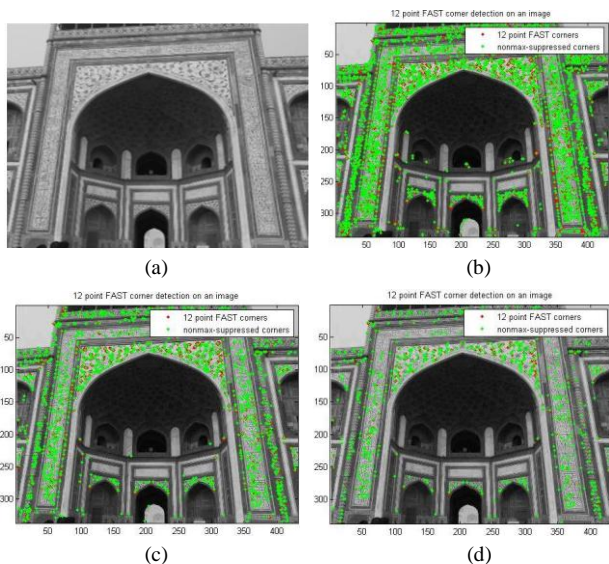
### A. SIFT Algorithm

The SIFT features were extracted and plotted on the image so as to get a detailed view of the keypoints. The SIFT algorithm is derived from its descriptors. Hence, the total time calculated for the extraction of features is based on the time of descriptor calculation. Fig. 5 shows the keypoints extracted from an image by applying the SIFT algorithm. Table 1 shows the simulation result parameters, such as the time required for the calculation of keypoints





**Fig. 5.** Simulation result of SIFT algorithm. (a) Original image, (b) image rotated by 45°, (c) straightened image, (d) keypoints of original image, and (e) keypoints of straightened image.



**Fig. 6.** Simulation result showing the corner points at varied threshold values. (a) Original image, (b) T = 20, (c) T = 30, and (d) T = 40.

and descriptors, intensity variation, rotation by angle, and different values of sigma. We have used  $\sigma = 1.6$  which provides optimal repeatability and maximum efficiency in terms of image matching. The larger the number of keypoints matched, the more efficient will be the object recognition.

**Table 1.** Parameters obtained from scale invariant feature transform (SIFT) algorithm

Method	Keypoints extracted	Time for finding keypoints (s)	Time to calculate descriptors (s)	Sigma ( $\sigma$ ) value	Matching (%)
SIFT	2169	0.116	2.914	1.6	100

**Table 2.** Parameters obtained from features from accelerated segment test (FAST) algorithm

Method	Threshold	Number of corner points detected	Time for finding corner points (s)
FAST	20	3913	0.716
	30	1714	0.651
	40	784	0.703

### B. FAST Algorithm

A machine learning algorithm was employed in a FAST-12 detector which detected corner points with a higher speed and repeatability. A non-maximal suppression algorithm is applied to the detector to remove the adjacent corners. The simulation result shows (Fig. 6) that the FAST-12 detector has dramatic improvements in repeatability over FAST-9 in noisy images. We focused our attention to optimize the detector for computational efficiency and make it more useful for real-time applications such as object recognition.

The figure was obtained with different threshold values and was observed that the number of corner points detected varied in accordance with the threshold set as shown in Table 2. From Table 2, it is found that at a threshold of 30, the corner points are detected in the image with less computational time than the threshold values of 20 and 40.

## IV. CONCLUSION AND FUTURE WORK

In this study, we analysed and compared the results of SIFT and FAST detector algorithms for object detection based on corner and feature points with respect to efficiency, quality, and robustness. The overall performance shows that although SIFT keypoints are useful because of their distinctiveness, the FAST algorithm has the best average performance with respect to speed, number of keypoints, and repeatability error. This property of SIFT features enables correct matching for a keypoint with other keypoints from a large database. Therefore, the SIFT algorithm has better performance in object detection although it is not improved for real-time applications. However, the machine learning approach used in FAST makes the algorithm faster in computation which adds to improvement in video quality, unlike other algorithms. FAST could achieve real-time performance in the object detection in an embedded device with increased speed.

However, the overall performance of FAST for object detection is not significantly high as compared to other object detection methods because this algorithm is a little insensitive to orientation and illumination changes. Therefore, it is concluded that the FAST algorithm performs better with higher computational speed and is suitable for real-time applications. Future work in this area will focus on modifying the FAST algorithm for better feature detection with accurate orientation and responsive to changes in illumination while maintaining the processing speed.

## REFERENCES

- [ 1 ] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, Nov. 2004.
- [ 2 ] K. Jeong and H. Moon, "Object detection using FAST corner detector based on smartphone platforms," in *Proceedings of the 1st ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering (CNSI)*, Jeju, Korea, pp. 111-115, 2011.
- [ 3 ] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, pp. 2567-2571, 2011.
- [ 4 ] C. Arth, C. Leistner, and H. Bischof, "Robust local features and their application in self-calibration and object recognition on embedded systems," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR2007)*, Minneapolis, MN, pp. 1-8, 2007.
- [ 5 ] E. Rosten and T. Drummond, "Machine learning for high speed corner detection," in *Proceedings of the 9th European Conference on Computer Vision*, Graz, Austria, pp. 430-443, 2006.
- [ 6 ] E. Rosten, R. Porter, and T. Drummond, "FASTER and better: a machine learning approach to corner detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105-119, 2010.
- [ 7 ] A. P. Witkin, "Scale-space filtering," in *Proceedings of the 8th International Joint Conference on Artificial Intelligence (IJCAI)*, Karlsruhe, Germany, pp. 1019-1022, 1983.
- [ 8 ] T. Lindeberg, "Generalized axiomatic scale-space theory," in *Advances in Imaging and Electron Physics*. Amsterdam: Elsevier Science, pp. 1-96, 2013.
- [ 9 ] M. Brown and D. G. Lowe, "Invariant features from interest point groups," in *Proceedings of the British Machine Vision Conference (BMVC)*, Cardiff, Wales, pp. 656-665, 2002.
- [10] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 530-534, 1997.
- [11] S. Edelman, N. Intrator, and T. Poggio, Complex cells and object recognition [Internet], Available: <http://kybele.psych.cornell.edu/~edelman/Archive/nips97.pdf>.
- [12] J. Beis, and D. G. Lowe, "Shape indexing using approximate nearest-neighbour search in high dimensional spaces," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Puerto Rico, pp. 1000-1006, 1997.
- [13] P. V. C. Hough, "Method and means for recognizing complex patterns," U.S. Patent 3069654, 1962.
- [14] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary patterns," *Pattern Recognition*, vol. 13, no. 2, pp. 111-122, 1981.
- [15] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV)*, Beijing, China, pp. 1508-1511, 2005.
- [16] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81-106, 1986.
- [17] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 3, pp. 660-674, 1991.



### Arpita Mohapatra

received her B.Tech in Electronics & Tele-communication engineering from Konark Institute of Science & Technology, in 2010 and M.Tech in VLSI & Embedded System from Institute of Technical Education & Research, SOA University, Odisha, India in 2013. Her research interest is video, image processing and VLSI system design.



### Sunita Sarangi

received her B.Tech degree in Instrumentation & Electronics Engineering from College of Engineering & Technology, India in 2001, M.Tech in Communication System Engineering from University College of Engineering, Burla, India in 2007. Presently working as assistant professor in dept. of Electronics & Instrumentation Engineering, Institute of Technical Education & Research, Siksha 'O' Anusandhan University, Odisha, India. She has over 9 years of experience in teaching and research. She has published research papers in journals and conferences. Her research interest includes signal and image processing special in biomedical image processing.



**Sukanta Sabut**

received his Diploma in Rehabilitation Engineering from National Institute of Rehabilitation Training & Research, Cuttack, India in 1993, B.E. degree in Electronics and Telecommunication Engineering from Institute of Engineers (India) in 2001, M. Tech degree in Biomedical Instrumentation from VTU, Karnataka, India in 2005 and Ph.D degree from School of Medical Science & Technology from IIT Kharagpur in 2010. Presently working as associate professor in the Dept Electronics & Instrumentation Engineering, Institute of Technical Education & Research, SOA University, India. He has over 15 years of experience in both teaching and research. His research interest includes biomedical signal & image processing, neural prosthesis, rehabilitation engineering and biomedical instrumentation. He has published research papers in international journals and conferences. He is a member of IEEE, IFESS, and Rehabilitation council of India, Biomedical Engineering Society of India and the Institution of Engineers (India).



**Srikanta Patnaik**

received his Ph.D (Engineering) on Machine Intelligence from Jadavpur University in 1999. Presently working as professor in the Dept of Computer Science & Engineering, SOA University, Bhubaneswar, India. He has supervised 12 Ph. D theses and more than 30 M. Tech theses in the area of Machine Intelligence, Soft Computing Applications and Re-Engineering. Dr. Patnaik has published around 60 technical papers in international journals and conference. He is author of 2 text books and edited 12 books and few invited book chapters, published by leading international publisher like Springer-Verlag, Kluwer Academic, etc. Dr. Patnaik was the Principal Investigator of AICTE sponsored TAPTEC project “Building Cognition for Intelligent Robot” and UGC sponsored Major Research Project “Machine Learning and Perception using Cognition Methods”. He is the Editors-in-chief of International Journal of Information and Communication Technology and International Journal of Computational Vision and Robotics published from Inderscience Publishing House, England and also Editors-in-chief of Book Series on “Modeling and Optimization in Science and Technology” published from Springer, Germany.