

## Research Article

# FAB: Fast Angular Binary Descriptor for Matching Corner Points in Video Imagery

**Mehwish Tahir, Nadia Kanwal, and Fatima Anjum**

*Department of Computer Science, Lahore College for Women University, Jail Road, Lahore 54000, Pakistan*

Correspondence should be addressed to Nadia Kanwal; [nadia.kanwal@lcwu.edu.pk](mailto:nadia.kanwal@lcwu.edu.pk)

Received 26 November 2015; Accepted 8 May 2016

Academic Editor: Farrokh Janabi-Sharifi

Copyright © 2016 Mehwish Tahir et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Image matching is a fundamental step in several computer vision applications where the requirement is fast, accurate, and robust matching of images in the presence of different transformations. Detection and more importantly description of low-level image features proved to be a more appropriate choice for this purpose, such as edges, corners, or blobs. Modern descriptors use binary values to store neighbourhood information of feature points for matching because binary descriptors are fast to compute and match. This paper proposes a descriptor called Fast Angular Binary (FAB) descriptor that illustrates the neighbourhood of a corner point using a binary vector. It is different from conventional descriptors because of selecting only the useful neighbourhood of corner point instead of the whole circular area of specific radius. The descriptor uses the angle of corner points to reduce the search space and increase the probability of finding an accurate match using binary descriptor. Experiments show that FAB descriptor's performance is good, but the calculation and matching time is significantly less than BRIEF, the best known binary descriptor, and AMIE, a descriptor that uses entropy and average intensities of informative part of a corner point for the description.

## 1. Introduction

To build an autonomous computing device, computers are required to achieve human vision. This includes detecting and recognizing different objects [1], segmenting objects from their background [2], classifying different scene segments for identification or stitching [3], text classification and identification, and face or hand recognition [4]. Accurate matching of image content is the fundamental part of all these vision based tasks, for which identification and description of salient image regions are the method that is closely related to human vision. Therefore, for the last two or more decades a number of detection and description techniques have been developed. The most popular of these are local image feature extraction and description techniques, which include edges, corners, blobs, or ridges as low-level image features.

Recognizing only distinctive image areas such as corners or blobs is not enough because they cannot be directly matched in other transformed images such as rotated, scaled, or illuminated. However, if we can get sufficient neighbourhood information on a feature, then it becomes easier

to identify the same image point in different transformed images. The collection of this neighbourhood information is called description. Although all types of image features are important, corner points are more informative [5]. This is because they carry some inherent information as compared to other features such as edges or blobs. An edge can occur independently anywhere in the image where there is some intensity difference [6]; similarly a blob is a sovereign image point identifying significantly bright or dark area with respect to its neighbourhood [7, 8], which can be matched. However, corner points can only appear when there are two or more intersecting edges. The presence of two or more edges gives additional information such as the presence of an object or existence of two separate planes which can be used for image contents' classification. The literature shows a number of feature descriptors; however, none of them explore this inherent information available with the corner points.

Previously proposed state-of-the-art descriptors use image pixels gradient and its orientation such as Scale Invariant Feature Transform (SIFT) [9], Speeded Up Robust Features (SURF) [10], and Gradient Location Orientation

Histogram (GLOH) and store it as numeric/float descriptor data for matching. These descriptors show very good performance in matching images under transformations, but at high computational cost. However, some other descriptors use binary vectors to store distinctive gradient information of a feature's neighbourhood such as Binary Robust Independent Elementary Features (BRIEF) [11], Oriented Fast and Rotated Brief (ORB) [12], Binary Robust Invariant Scalable Keypoints (BRISK) [13], and Fast Retina Keypoint (FREAK) [14]. Binary descriptors gained more attention due to being less computationally expensive yet considerably accurate. Although numeric descriptors are more accurate than binary descriptors, they need more time to compute and match which make them undesirable for real-time applications. Furthermore, almost all of the aforementioned descriptors describe circular area around feature point for description and, therefore, produce large size descriptors.

This paper presents a descriptor that describes the useful neighbourhood of a corner point: in between two edges and hence named Fast Angular Binary (FAB) descriptor. It combines the strengths of two descriptors BRIEF and a recently proposed numeric descriptor for corner points called Angle, Mean Intensity and Entropy of Informative Arcs (AMIE) [15]. BRIEF takes the circular region around feature point to create binary descriptor and, therefore, cannot produce good matches in dynamically changing backgrounds, whereas AMIE selects only the informative area (that lies inside two edges) of a corner point but computes descriptor using basic image intensities and hence produces more false matches. FAB on the other hand combines the distinctiveness of BRIEF and efficiency of AMIE to calculate more capable, robust, and reliable descriptor. Therefore, it is more suitable for identifying objects with dynamically changing backgrounds in real-time. Contrary to BRIEF and AMIE, FAB is capable of accommodating scale and rotational changes while matching frames of a video for tracking, robot navigation, and other video analysis based applications. Furthermore, minimum processing should enable faster processing of data in autonomous robot navigation and other applications.

The rest of the paper is organized as follows: Section 2.1 describes the working principles of BRIEF, AMIE, and FAB. Sections 2.2 and 3 present the matching of FAB descriptor and how it is more efficient and accurate than other descriptors. Performance of newly developed descriptor is compared with BRIEF and AMIE in Section 4.1. Section 4.2 describes the application of matching images using FAB descriptor in different vision applications and lastly Section 5 concludes the paper and suggests some future modification.

## 2. Material and Methods

**2.1. Feature Descriptors.** Feature description is a process of creating vectors of neighbourhood information of a feature point. This descriptor is supposed to contain sufficiently unique information to be matched in another transformed image. These transformations include scaling, rotation, illumination, and change in viewpoint also called affine transformation. However, considering a problem of object tracking in video imagery, the amount of image transformation is quite

low. For example, at a normal frame rate the scale changes between frames are very small and sometimes negligible. Similarly rotation only occurs due to camera jittering and there is no affine transformation in consecutive video frames. Therefore, for video imagery feature, matching does not need complex computation to make descriptors scale, rotation, or affine invariant which happens in case of SIFT and SURF. Moreover, small and binary descriptor vectors take considerably less time to compute and match and, therefore, are more attractive for real-time applications such as BRIEF and AMIE. The following sections briefly describe their working principles.

**2.1.1. BRIEF.** It randomly selects two pixels  $P_i$  and  $P_j$  from a circular neighbourhood of a feature point. The intensities of these two pixels are compared to produce a binary value (0/1) as shown below:

$$\tau(p; x, y) = \begin{cases} 1 & \text{if } P_i(x) < P_j(y); \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

A 64-bit descriptor vector is produced for each image feature. But due to the binary nature of the descriptor, matching becomes efficient—calculating Exclusive OR, Jackard-Needham, or Dice matrix [16] of two vectors is very fast. However, for BRIEF, the selection of pair of neighbourhood pixels to be compared is an important step. Authors in [11] performed five tests to select the best sample pair and found Isotropic Gaussian distribution to be able to produce good number of matched points as compared to the selection of pixels according to some pattern such as equivalent distance points.

**2.1.2. AMIE.** AMIE is a numeric descriptor that stores mean intensity and entropy of circular arcs around a corner point. Furthermore, these arcs only cover the informative region around corner point instead of whole circular area. Although the descriptor contains very basic pixels information, it becomes discriminating when calculated only for arc pixels. It produces a vector of length of 16 only that contains angle, mean intensity, and entropy of different arcs and a value to describe the direction of informative region. Although AMIE is a numeric descriptor, its matching time is comparable to that of BRIEF due to its small size. Likewise, AMIE is not scale invariant and is not suitable to be used in applications where matching is required between transformed images such as scaled, rotated, or illuminated.

**2.1.3. FAB.** Fast Angular Binary (FAB) descriptor for corner points is a descriptor that combines the strengths of BRIEF and AMIE discussed above to facilitate matching in real-time with improved accuracy. It calculates the binary vector for five different arcs of different radii similar to AMIE. However, its descriptor is completely different from AMIE because of its variable length vectors depending on corner's angle. So every corner point may have different length descriptor; however, an average length of 60 bits is found for 1000 video frames. This includes binary vector for five different arcs of radii 3,

5, 7, 9, and 11 along with angle and a direction bit. AMIE's descriptor's length is fixed as it stores collective information of each arc such as average intensity and entropy. Detailed description of steps required to calculate FAB descriptor is given below.

Following the detection of corner points in an image using a good detector (Harris and Stephens algorithm [17] is used here), below is the pseudo code followed for descriptor calculation:

```

Read video frames
for each frame
{
  (1) convert into grayscale
  (2) detect corner points
  (3) for all corner points
  (4) find edges around the corner
  (5) calculate angle between the two detected edges
  (6) find orientation of the angle
  (7) select orientation most related to corner point
  (8) calculate descriptor for only selected part of the corner
}

```

The following sections describe these steps in detail.

**2.1.4. Angle and Orientation of a Corner Point.** Figure 1 shows five circular arcs of radii 3, 5, 7, 9, and 11 around a corner point. These radii are selected to cover the maximum nonoverlapped neighbourhood area. Further, these circular arcs are used to find edge pixels which then decide the angle of a corner point. Each circular arc is scanned to find eigen values ( $\lambda_1$  and  $\lambda_2$ ) for the categorization of image pixels according to Figure 2. Angle of the corner points is calculated by counting number of pixels between two edge pixels at each arc as explained by (2), where  $N_r$  is the number of pixels in an arc of radius  $r$  while  $M_r$  is the number of pixels in the circle of radius  $r$ . Consider the following:

$$\theta_r = 2\pi \left( \frac{N_r}{M_r} \right). \quad (2)$$

Although due to rasterization in digital images each arc can give different angle information, it is useful at the same time as it compensates the irregular shapes (nongeometrical shapes such as hands and clothes). The average angle of all five arcs is stored as corner's angle. Furthermore, these edge pixels divide the circular area around corner point into two parts. One may give information about an object (maybe from edge 1 to edge 2 in Figure 1) and the other one may belong to background (from edge 2 to edge 1 in Figure 1).

Once we have angle information next step is to find the orientation of its internal part which should be used for the description purpose. Logically the part which shows more similarity to the corner point should be selected. Therefore the intensities of each part are compared with corner point

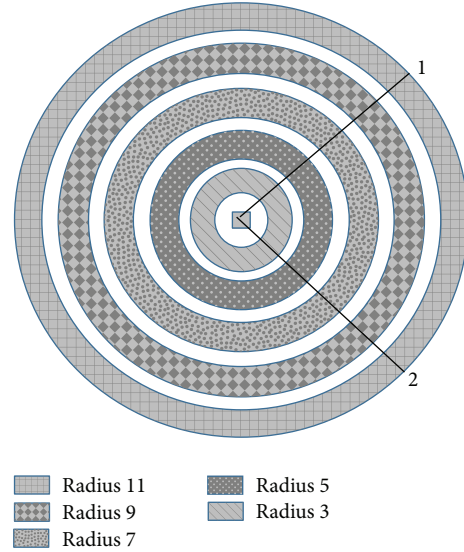


FIGURE 1: Corner point, corresponding edges, and 5 radii around it used to identify internal and external part for description.

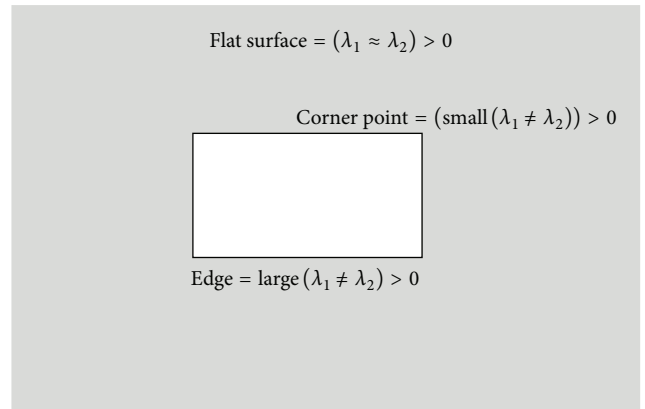


FIGURE 2: Eigen values of image pixels used to classify pixels as corner, edge, or representing flat surface.

and the one that shows close average intensity is selected for descriptor calculation and its direction is stored as the orientation of a corner's angle. For simplicity bit value 1 represents clockwise direction and 0 represents anticlockwise direction.

**2.1.5. Descriptor Calculation.** Conventionally, binary descriptors are constructed by comparing either randomly selected pair of pixels or some specified patterns. For example, BRIEF choose pixel pairs based on Isotropic Gaussian distribution while FREAK uses a pattern similar to human eye retina and it claims to give better performance than BRIEF. Beneficially, selecting circular arcs gives patterns similar to FREAK and, therefore, appears to be more effective. Each arc of different radius contributes to the descriptor data. Because of dissimilar radii the number of bits for each arc is different which is why it is called variable length descriptor. Maximum number

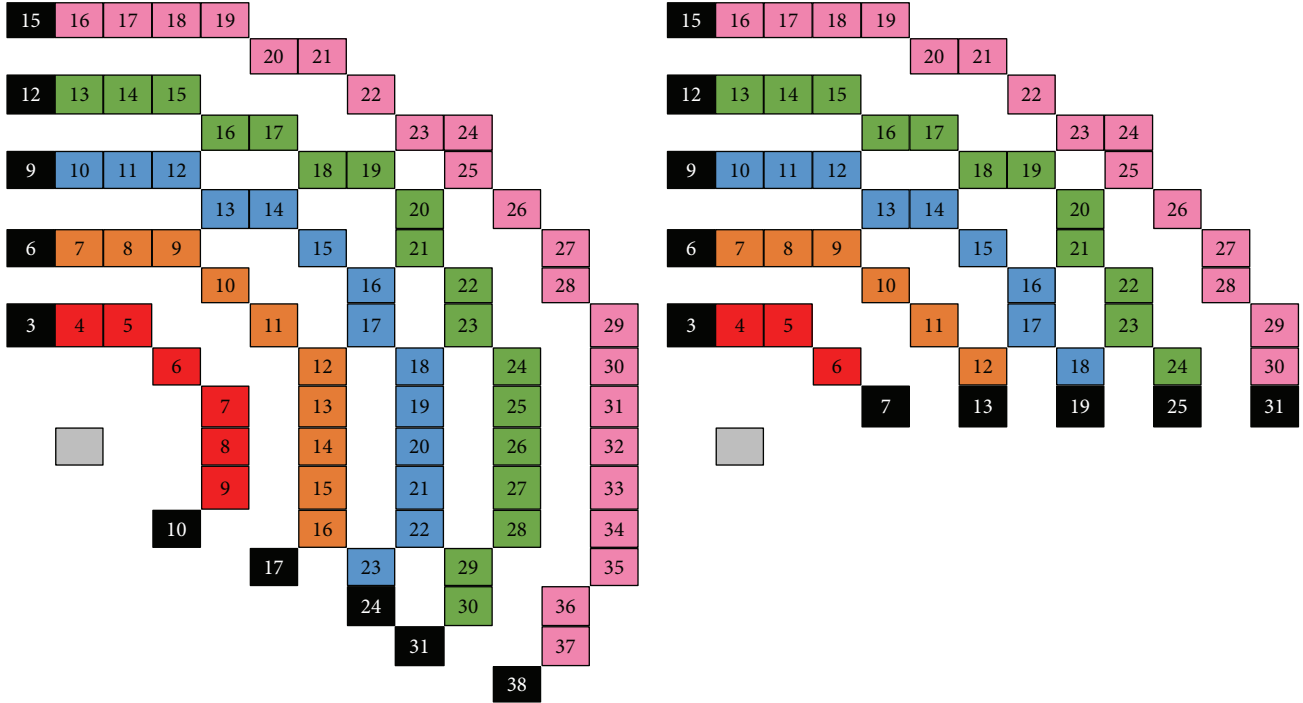


FIGURE 3: Two corner points with different angles and, therefore, different arc lengths for description.

of bits for each arc is equal to the total number of pixels in the circle of radius  $r$ , which are used for descriptor calculation, although, rarely, it happens when there is only one edge pixel found on majority of the circular arcs or no edge pixel is found. Describing this kind of detected points can be argued because they should not be detected as corner points and can be considered false responses of corner detection algorithm. However, to increase the amount of information about whole image content we considered them equally important and complete circular arcs are used for the description purpose. In order to build binary descriptor for the internal part of corner point, consecutive pixels at each arc are compared using the following equation:

$$\tau(p; x, y) = \begin{cases} 0 & \text{if } \text{abs}(p(x) - p(y)) < t; \\ 1 & \text{otherwise,} \end{cases} \quad (3)$$

where  $t$  is a threshold set to 1.5 here. Figure 3 shows two corner points with different angles which usefully resulted in different size of arcs or descriptor. For left side corner point the number of pixels is 6, 10, 14, 18, and 22 for radii 3, 5, 7, 9, and 11, respectively. So the descriptor length will be 71; similarly the descriptor length of right side corner point is 46 including 6 bits of each arc's angle and one bit of orientation bit, smaller than BRIEF or FREAK (64 bits) and SIFT or SURF (128 bits).

Edge pixels are not used in descriptor calculation because of maximum intensity difference from its neighbouring pixels. Therefore, it does not contribute in making descriptor more discriminative.

**2.2. Matching Corner Points.** Matching is the step that makes FAB most attractive descriptor. As discussed before the FAB descriptor contains numeric and binary values purposefully. Therefore, matching is performed in two steps that are

- (i) matching angle and orientation,
- (ii) matching binary descriptor.

This two-step matching technique reduces the search space and helps finding optimal solution all the time. Binary descriptor only matched the corner points for which the orientation is the same and the angle is in between some specified range. This further helps achieving scale invariance for FAB descriptor.

Orientation and angle part of the descriptor are matched by finding minimum Euclidean distance of each corner point of reference image from all corner points belonging to test image. So every time a close match is found, binary descriptor is matched using XOR and its hamming distance is stored as matching difference which should be minimum. Hence, at each iteration if the previous match difference is greater than the newly calculated one, current point is selected as a better choice and its difference is stored for future comparisons. This way it is possible that, comparing orientation and angle part of one point with 500 points, the binary descriptor may match only 10 of these 500, a more than significant reduction of unnecessary comparisons. However, it is important to highlight the significance of FAB's variable length. If the two descriptors under comparison have different length, then both descriptors are aligned in such a way that ensures correct matching part. For this the central

part of both descriptors is aligned leaving left and right bits of larger descriptor unmatched. Although there seems to be some loss of information, practically it is not the case because of matching corner points with similar angles; maximum of two to four bits are usually unmatched. Further it facilitates matching images under scale transformation making FAB scale invariant. Experimental results shown in next sections prove this argument.

**2.2.1. Outliers Removal Using RANSAC.** Matches are computed until this step can have some wrong matches. This is because descriptors of a number of points in the image can have similar angles at different locations. Therefore, to discard wrong matches, Random Sample Consensus (RANSAC) method is used. This method finds a homography matrix (a  $3 \times 3$  matrix representing translation and rotation between an image pair) between two images using a randomly selected subset of matched points. Then, it works out support in favour of calculated homography from the rest of the matches and keeps the homography with maximum support. The matched points favouring the selected homography are considered as true matches and also called inliers and the rest as false matches or outliers.

### 3. Computation and Matching Time Comparison

Figure 4 shows average computation time of three descriptors for 100 images containing approximately 50,000 corner points. Detection of corner points is done using Harris and Stephens corner detector for all three descriptors; therefore, its computation time remains the same. Descriptor computation time of FAB and AMIE is slightly more than BRIEF, but still less than 0.2 seconds. However, significantly less matching time of FAB descriptor makes it an overall winner. It is definitely due to angle based matching of binary descriptor, which itself is very fast.

### 4. Results and Discussion

**4.1. Performance Evaluation.** Performance of FAB descriptor is compared with BRIEF and AMIE descriptors. This descriptor is mainly developed for matching video frames for applications such as object tracking, template matching, and navigation. Therefore, a number of video results are presented here. As it is not possible to show thousands of video frames as images in the paper, a framework is defined for this purpose called as difference of images (DOI). Computation of DOI is done using the following procedure:

- (i) Find homography for pair of images using corresponding matched points.
- (ii) Warp the first frame using the computed homography matrix.
- (iii) Subtract warped frame from second frame and result will be zero image in case of good matches and a nonzero image otherwise. In other words the sum of pixel intensities DOI will be close to zero if the image

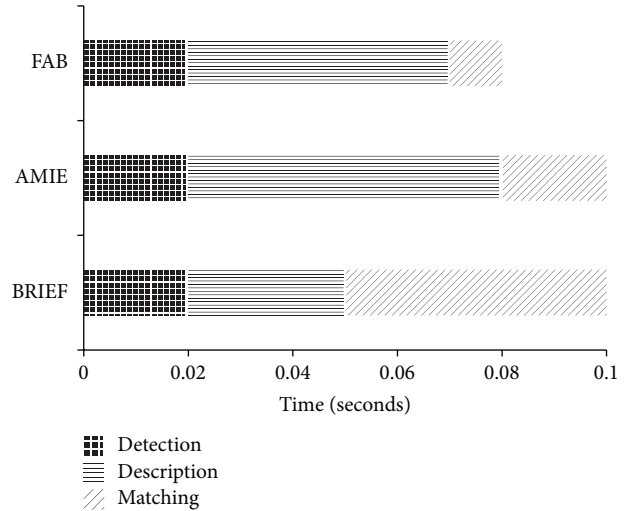


FIGURE 4: Comparison of computation and matching time of FAB with BRIEF and AMIE averaged over 100 images.

is warped using good matched points and greater than zero otherwise.

This framework is applied in all of the performance evaluation tests presented below. For testing a number of videos are captured using handheld camera along with using surveillance video data publicly available from PETS2009 [18]. Furthermore, homography matrix has been previously used and proved to be a better performance analysis metric as compared to repeatability or detection rate [19]. Before discussing these results, it is worth showing that FAB is more scale invariant than AMIE or BRIEF. For this purpose three tests have been performed using two different images and a video sequence, in which images with scale difference are matched and results are shown below.

Figure 5 shows such images and Figure 6 displays the result of FAB descriptors while matching each reference image with corresponding scaled images. Similarly, to show matching results under scale change in video imagery, first frame of a video captured by a handheld camera is matched with subsequent frames as shown in Figure 7. Here, two matched pairs of frames are shown and because video matching results are difficult to display in printed form, a graph is shown in Figure 8 where the sum of all pixel intensities in difference of warped images (DOI) is plotted for all three descriptors. In case of good results the DOI should contain black pixels (with 0 pixels' intensities) for which the sum of whole image will be close to zero in the graph. The results produced by FAB descriptor shown in Figure 8 outperformed all others by producing good matched points and minimum value of DOI (sum of pixel intensities of difference image) for whole video.

**4.2. Effectiveness of FAB in Different Vision Applications.** Performance evaluation of FAB descriptors for scaled images shows promising results both in still images and in video imagery. Therefore, it should also be tested for different vision

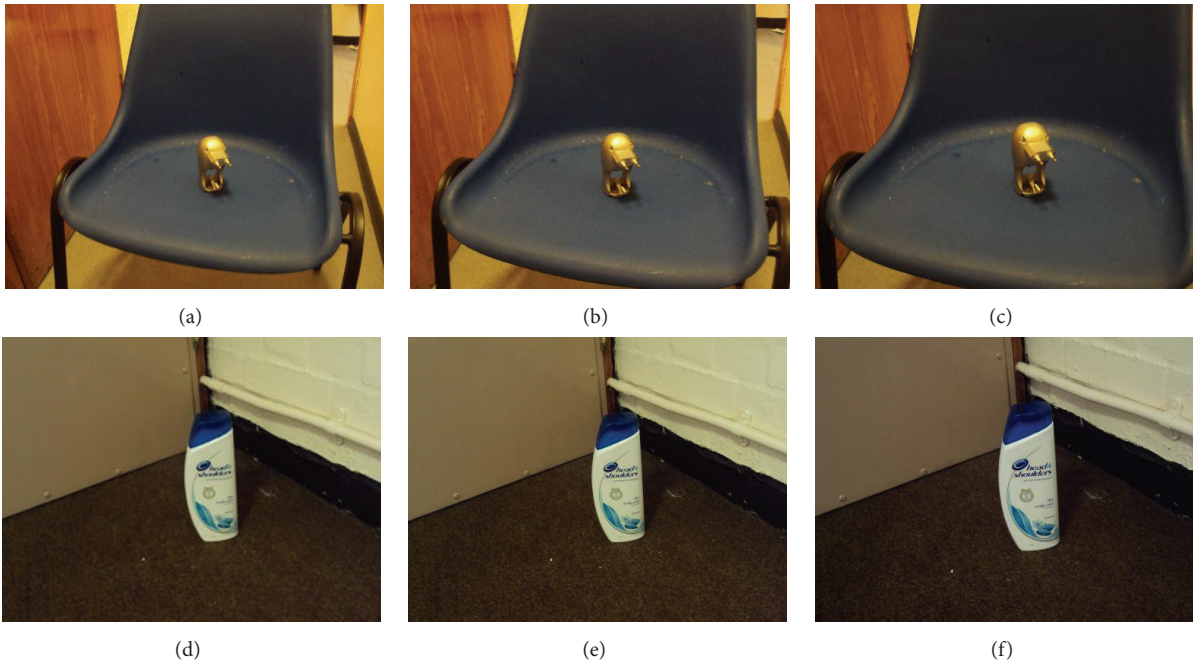


FIGURE 5: Images with scale differences. First image in each row is selected as reference image and matched with test images (second and third in the row).

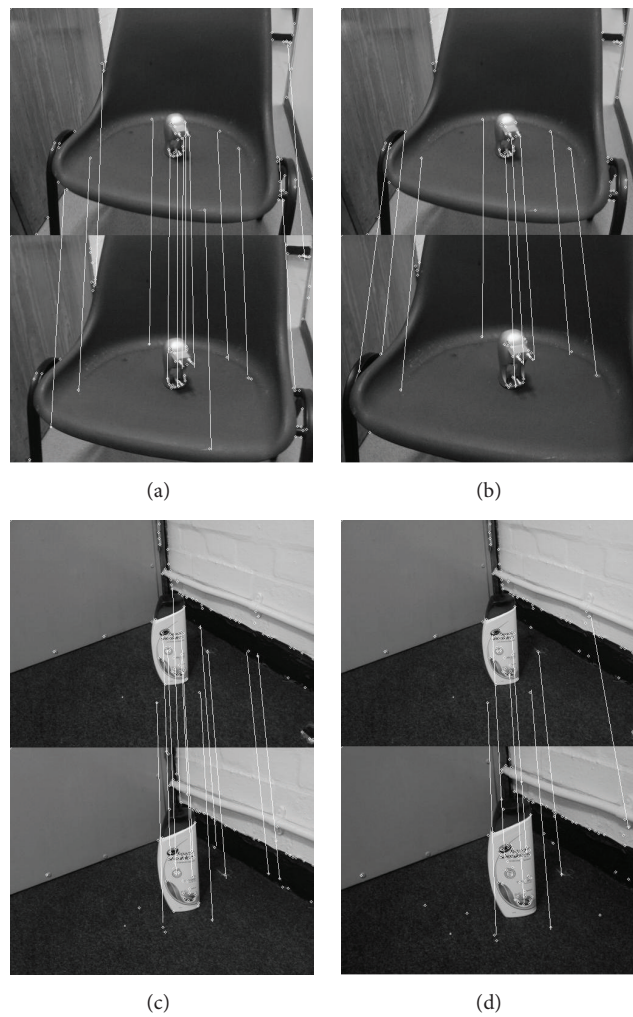


FIGURE 6: Scaled image pairs matched using FAB descriptor. Reference image in Figure 5 is matched with corresponding test images.

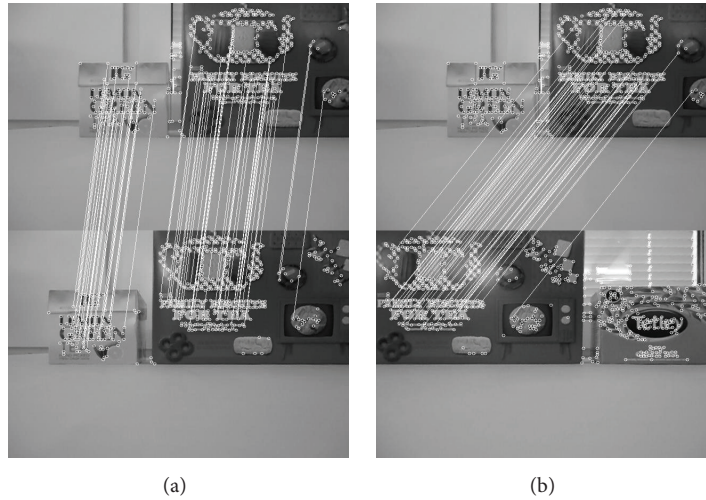


FIGURE 7: Initial video frame is matched with subsequent frames of the video using FAB descriptor to show matching under scale differences. Results are shown for frame #1 matched with frame #150 (a) and frame #205 (b).

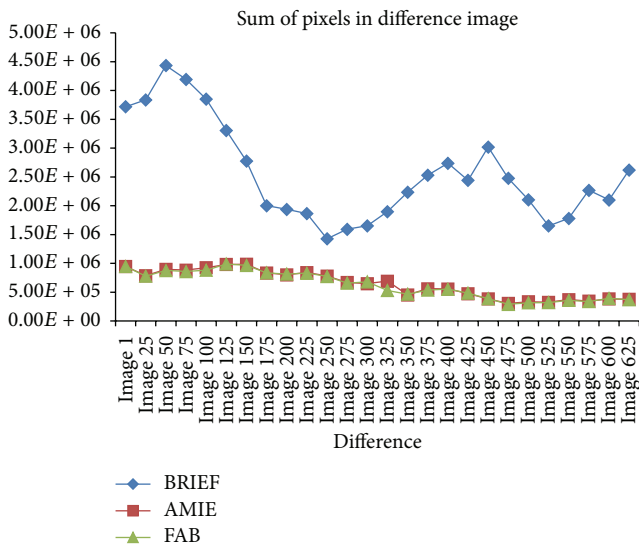


FIGURE 8: DOI analysis of video frames captured using handheld moving camera (as shown in Figure 7); small values show minimum difference in matched images and hence good matching results.

applications such as template matching, motion detection in surveillance videos, and object tracking.

**4.2.1. Object Tracking.** Object tracking is one of the most popular applications of computer vision. Figure 9 shows two sample frames from a video captured using a handheld camera. In the video a car is moving and its position is tracked by finding corner points which are matched in subsequent frames using FAB descriptor. Matching results obtained by FAB descriptor to track this moving car are presented in Figure 10. The tracked car's position is marked by a rectangle in the difference images.

**4.2.2. Surveillance.** Next application, that is, motion detection using surveillance cameras, has great significance in today's world where security is becoming the foremost global and local issue. Vision based surveillance systems are getting more attention these days because of its low cost and an unobstructed use of sensor (camera/s). PETS is a very popular public domain for vision community where a number of surveillance video datasets are available for research purposes. We also took advantage of getting different surveillance videos to test FAB descriptor. Some sample images are shown in Figure 11 while the matching results are shown in Figure 12 where the movement can easily be detected using DOI.

**4.2.3. Template Matching.** Lastly, template matching is done to show that FAB can work equally good for applications such as vision based engineering, medical, and text classification. A template containing some text is matched in a video sequence. Figure 13 shows the template which is matched in different video frames. Figure 14 shows accurately matched template in different timed spaced video frames. In this figure first two frames are initial part of the video while last two frames are from the middle to end part of the video.

FAB descriptor for corner points proposes improvement in two different dimensions, one using corner points in different vision applications such as tracking, surveillance, and template matching which helps identify useful image content. Secondly, it describes only the useful neighbourhood of corner point using binary vectors which facilitate matching an object in changing background as can be seen in tracking and template matching results shown in Figures 10 and 14. Moreover, these results along with matching results of scaled images shown in Figures 6 and 7 proved FAB to be scale invariant descriptor that is very useful and required property of a fast feature descriptor. Time comparison of FAB with



FIGURE 9: Sample frames from a video captured using a handheld camera with a moving car.



FIGURE 10: Tracking a moving car using matched corner points by FAB descriptor; DOI are showing the new location of the moving object.





FIGURE 11: Sample frames from a PETS2009 dataset View\_001 captured using a fixed camera.

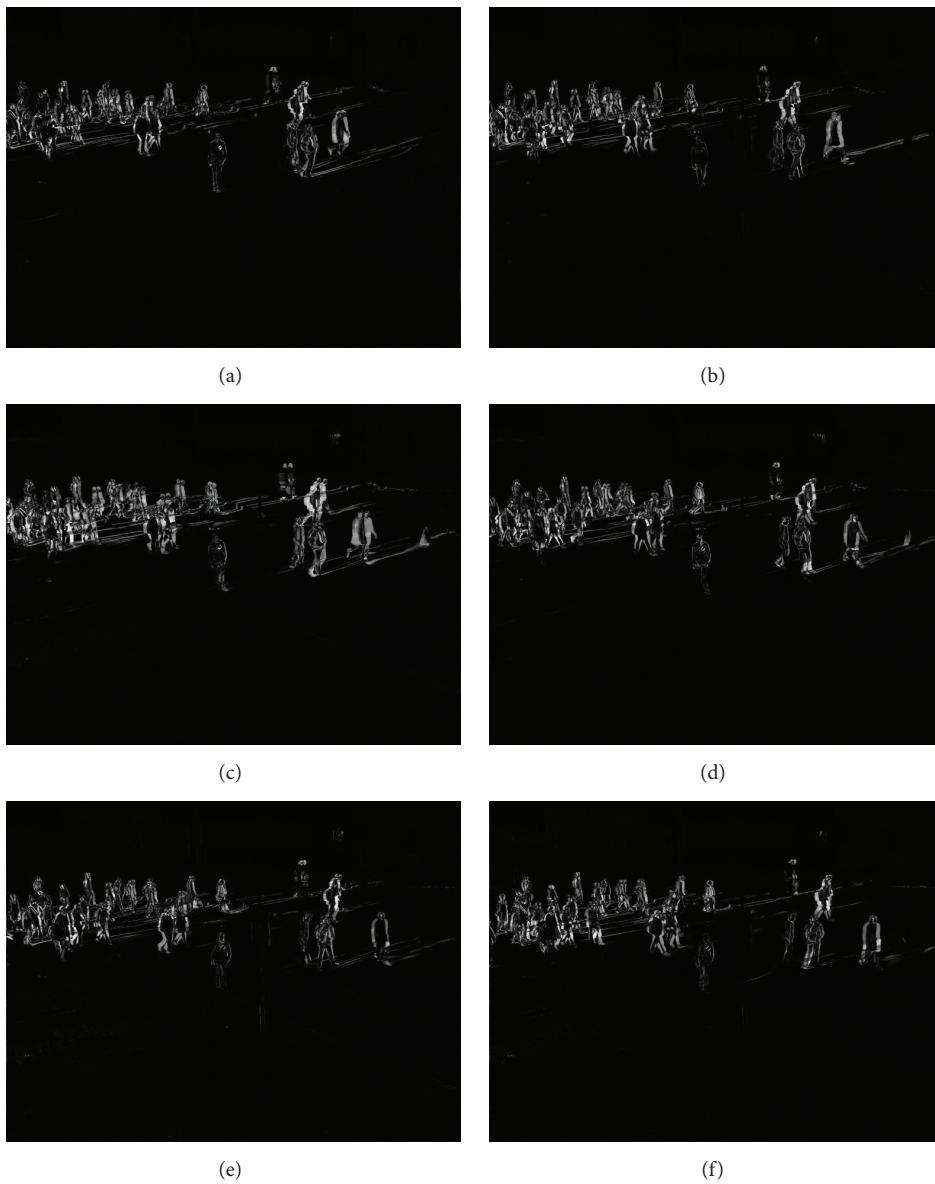


FIGURE 12: PETS dataset is used for motion detection using surveillance cameras. Dataset used in this application is captured using still camera whereas objects are moving. The white area is showing that multiple persons are passing from this area. This application can be used to track if someone entered a place where entrance is not allowed.

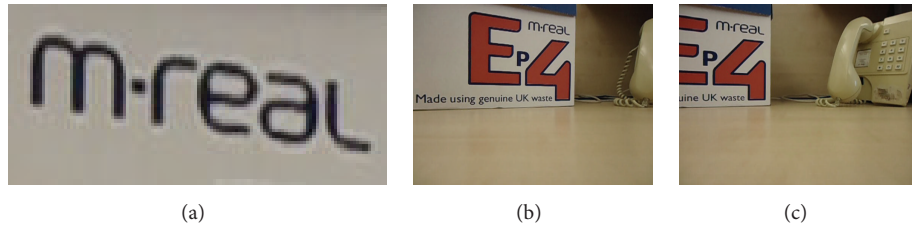


FIGURE 13: A template and some sample video frames for template matching. A moving handheld camera is used to capture this video in which a text template is matched.

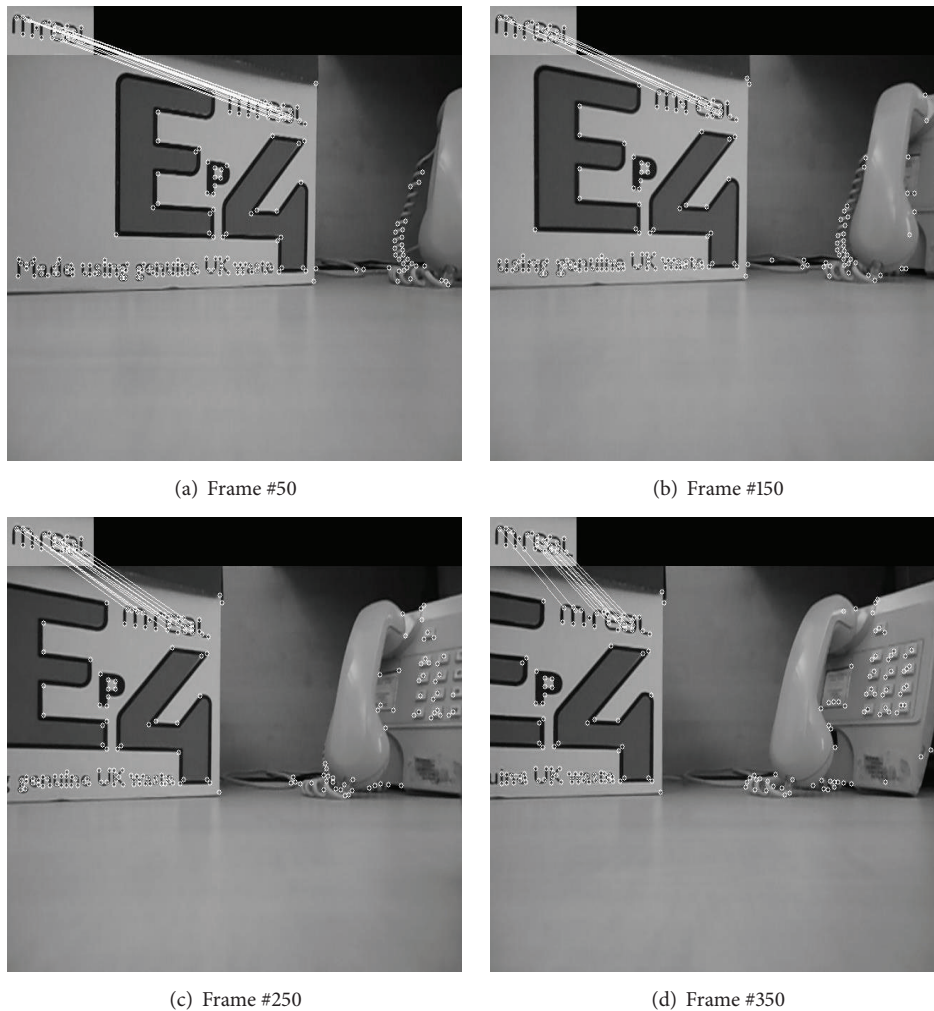


FIGURE 14: Template is successfully matched in whole video; however, some equally spaced frames are shown here.

BRIEF and AIME shows that it can be used in real-time applications with improved accuracy.

## 5. Conclusion

Efficient and robust feature descriptor is the key to match images accurately. Feature descriptor with less computation time and less storage space is the basic need of real-time

applications. This paper presented a new hybrid descriptor called FAB (Fast Angular Binary) descriptor, containing two kinds of feature's neighbourhood information, first angle and orientation of a corner point and the second binary vector representing pixels' intensity comparison for only useful neighbours of corner point. This hybrid information assists in implementing a two-step matching technique which is actually the trivial part of FAB's success. Matching only those

corner points which have similar angle and orientation makes search space significantly small and supportive for getting optimal results quickly.

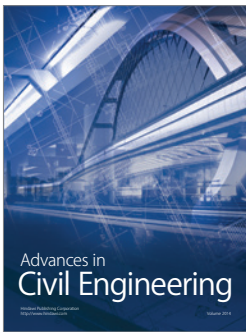
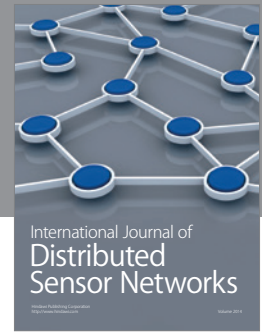
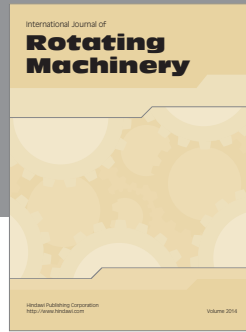
Performance comparison of FAB with other state-of-the-art descriptors proved it to be more efficient, reliable, and accurate for different vision applications. For future modifications, FAB descriptor can lose orientation bit to achieve rotation and affine invariance, but with some added information.

## Competing Interests

The authors declare that they have no competing interests.

## References

- [1] F. Porikli and A. Yilmaz, "Object detection and tracking" in *Video Analytics for Business Intelligence*, vol. 409 of *Studies in Computational Intelligence*, pp. 3–41, Springer, New York, NY, USA, 2012.
- [2] P. Kapsalas and S. Kollias, "Affine morphological Shape Stable Boundary Regions (SSBR) for image representation," in *Proceedings of the 18th IEEE International Conference on Image Processing (ICIP '11)*, pp. 3381–3384, Brussels, Belgium, September 2011.
- [3] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision*, vol. 74, no. 1, pp. 59–73, 2007.
- [4] R. R. Igorevich, P. Park, J. Choi, and D. Min, "Two hand gesture recognition using stereo camera," *International Journal of Computer and Electrical Engineering*, vol. 5, no. 1, pp. 69–72, 2013.
- [5] N. Kanwal, S. Ehsan, E. Bostanci, and A. F. Clark, "Evaluating the angular sensitivity of corner detectors," in *Proceedings of the IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems (VE-CIMS '11)*, pp. 28–31, Ottawa, Canada, September 2011.
- [6] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.
- [7] A. J. Danker and A. Rosenfeld, "Blob detection by relaxation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 3, no. 1, pp. 79–92, 1981.
- [8] S. Hinz, "Fast and subpixel precise blob detection and attribution," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '05)*, vol. 3, pp. 457–460, September 2005.
- [9] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV'99)*, vol. 2, pp. 1150–1157, September 1999.
- [10] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: speeded up robust features," in *Computer Vision—ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds., vol. 3951 of *Lecture Notes in Computer Science*, pp. 404–417, Springer, New York, NY, USA, 2006.
- [11] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: binary robust independent elementary features," in *Computer Vision—ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part IV*, vol. 6314 of *Lecture Notes in Computer Science*, pp. 778–792, Springer, Berlin, Germany, 2010.
- [12] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 2564–2571, Barcelona, Spain, November 2011.
- [13] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary Robust invariant scalable keypoints," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 2548–2555, Barcelona, Spain, November 2011.
- [14] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: fast retina keypoint," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 510–517, IEEE, Providence, RI, USA, June 2012.
- [15] N. Kanwal, E. Bostanci, and A. F. Clark, "Matching corners using the informative arc," *IET Computer Vision*, vol. 8, no. 3, pp. 245–253, 2014.
- [16] E. Bostanci, "Is Hamming distance only way for matching binary image feature descriptors?" *Electronics Letters*, vol. 50, no. 11, pp. 806–808, 2014.
- [17] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, vol. 15, pp. 147–151, Manchester, UK, September 1988.
- [18] J. Ferryman and A. Shahrokni, "An overview of the pets 2009 challenge," in *Proceedings of the 11th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, Miami, Fla, USA, 2009.
- [19] E. Bostanci, N. Kanwal, and A. F. Clark, "Spatial statistics of image features for performance comparison," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 153–162, 2014.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

