

# Feature Based Correspondence: A Comparative Study on Image Matching Algorithms

Usman Muhammad Babri<sup>1</sup>, Munim Tanvir<sup>2</sup> and Khurram Khurshid<sup>3</sup>

<sup>1,2,3</sup>Electrical Engineering Department, Institute of Space Technology  
1, Islamabad Highway, Islamabad, Pakistan

**Abstract**—Image matching and recognition are the crux of computer vision and have a major part to play in everyday lives. From industrial robots to surveillance cameras, from autonomous vehicles to medical imaging and from missile guidance to space exploration vehicles computer vision and hence image matching is embedded in our lives. This communication presents a comparative study on the prevalent matching algorithms, addressing their restrictions and providing a criterion to define the level of efficiency likely to be expected from an algorithm. The study includes the feature detection and matching techniques used by these prevalent algorithms to allow a deeper insight. The chief aim of the study is to deliver a source of comprehensive reference for the researchers involved in image matching, regardless of specific applications.

**Keywords**—computer vision; image matching; image recognition; algorithm comparison; feature detection

## I. INTRODUCTION

The recent decade has experienced drastic improvements in the field of computer vision, including scene or object recognition, stereo correspondence and motion tracking. Matching images and finding correspondence between them has been a key application to computer vision. This paper is a comparative study evaluating the performance of prevalent image matching algorithms. The performance is assessed according to the results obtained from various criteria such as speed, occlusion, sensitivity, etc. The paper also focuses on the level of accuracy achievable by an algorithm depending on the image type. This prediction ability comes in handy under the circumstances where resources are limited.

Achieving highly reliable results is the ultimate goal of any image matching method. However, none of them has gained universal acceptance. The type of image and the variations in the images to be matched are key elements in the selection of a matching method, along with the scale (two features of images have diverse scales), occlusion (two objects that are spatially separated in 3D plane might be interfering with each other when projected in 2D), orientation (two images are rotated with regard to each other), object to be matched (if the object is a planar, textured or edgy object), clutter (conditions of the image background) and illumination (fluctuations in illumination in two features).

The present image matching algorithms perform adequately under the above described conditions but even the most prevalent algorithms have not gained total invariance to these problems. However, through a methodical testing program, the

efficiency of the image matching algorithms will be tested on diverse sets of images with a noticeable difference in texture, clutter, orientation and other factors. These results will be eventually used to contrast between the distinctiveness of the found features. Therefore, our purpose here is to weigh the performance of these algorithms under various conditions using impartial standards.

## II. RELATED WORK

The roots of image matching and feature detection date back to 1981 to the work of Moravec using a corner detector for stereo matching. Initially, its applications were confined to stereo and short range motion tracking, but the revolutionary work of Schmid and Mohr (1997) illustrated how local invariant feature matching could be applied to image recognition problems for matching feature against a large database of images. They used Harris corner detector (1988) to select interest points which permitted features to be accorded under random orientation change. The Harris corner detector, however, was sensitive to changes in image scale. This was overcome by Lowe (1999) who achieved scale invariance by extending the local feature approach and then further extending it making the local features invariant to full affine transformations (Brown and Lowe, 2002).

### A. Scale Invariant Features Transform (SIFT)

Shortly after presenting an algorithm on feature detection from textured images, Lowe gave an improvised version of his work under the publication “Scale Invariant Feature Transform (SIFT) algorithm” (Lowe, 2004). This presents a technique to identify unique invariant features which may be used for comparison between different angles, orientations or viewpoints of an object or a scene. Lowe’s approach is broken down into four key components:

1) *Scale space extrema detection: to recognize probable interest points that are invariant to orientation and scale by means of a difference-of-Gaussian function.*

2) *Key point localization: a thorough model is used to define location and scale at every candidate location and then, centered upon measures of their stability, key points are selected.*

3) *Orientation assignment: according to local image gradient directions one or more orientations are allocated to every key point location. This provides invariance to changes in orientation, scale and location as all future calculations will be made relative to the assigned ones.*

4) *Key point descriptor*: at the area around each key point at the designated scale the local image gradients are measured and converted into a representation that tolerates substantial levels of variation in illumination and local shape distortion.

These stages are implemented in a descending order and only the key points that are robust enough jump to the subsequent stage. However, this proved to be an expensive process and therefore an upgraded version of SIFT's descriptor was presented "Principle Component Analysis SIFT (PCA-SIFT)" (Ke and Sukthankar, 2004). However, it proved to be less distinctive than SIFT. Another noteworthy approach is presented by Lindeberg under the name of "Scale Invariant Feature Transform" (Lindeberg, 2012). This technique for feature detection in the SIFT operator can be perceived as a variant of a scale-adaptive blob detection method where the detected blobs with related scale levels are found from scale-space extrema of the scale-normalized Laplacian.

### B. Speeded Up Robust Features (SURF)

Shortly after the PCA-SIFT another image matching algorithm was put forward that was to ensure speed in: detection, description and matching. This algorithm was named Speeded-Up Robust Features (SURF) detector (Bay et al., 2006). Contrary to other prevalent approaches of the time, SURF uses hessian matrix to considerably increase the matching speed. It depends on integral images to lessen the computation time and its descriptor defines a scatter of Haar-wavelet responses around the interest point. With the low dimensionality descriptor (64-dimensions) and 'Fast-Hessian' detector SURF is certain to perform faster.

### C. Features from Accelerated Segment Test (FAST)

Created by Trajkovic and Hedley (1998), FAST is the only feature-based algorithm used for this comparison. However, the implementation used for the comparison was published by Edward Rosten (Rosten and Drummond, 2006). The FAST detector is a wedge type detector i.e. a corner is detected using a circle surrounding a candidate pixel. It operates by considering a circle of 16 pixels and if there happen to be  $n$  adjoining pixels are above or below a threshold value,  $t$ , then the candidate pixel is chosen to be a corner. However, Rosten and Drummond extended this algorithm to use a machine learning based detector. Where all other detectors identify corners using an algorithm this technique trains a classifier on the model and then apply the classifier to an image. The classifier can be trained on how a corner should behave. This

makes the detector to perform significantly faster than other feature detectors.

### D. Other comparative studies

Various people have published studies to compare different image matching algorithms. Few of the most recent of these are Comparative Study of Image Matching Algorithms (Babbar et al., 2010), A comparison of SIFT, PCA-SIFT, and SURF (Juan and Gwon, 2009) and A Comparative Study of Three Image Matching Algorithms: Sift, Surf, and Fast (Guerrero, 2011). They have been effective in relating most aspects of the algorithms. However, their work does not present us either with the pictures they have used or with the system specifications or only textured images being used for the comparison.

The study aims at complementing the already available studies by allowing the quick recognition of the process that will perform efficiently under the given specifications. The paper also focuses on investigating the accuracy of the algorithms in detecting a single object in a scene.

## III. METHODOLOGY

The existing approaches towards the evaluation of the prevalent algorithms depend on their performance when applied on similar datasets. This has directed to different conclusions where one of these algorithms is presented as the finest, while in another publication the same algorithm performed contrarily. In our opinion, the major reason for this is that the image matching algorithms are best suited to a particular type of image and they would perform better when tried on these particular types of images. The suggested study will be based on the comparison of different image datasets. Herewith the hypothesis declaring that the performance of an algorithm depends on the type of image dataset.

Three image datasets will be taken under consideration during the study to compare the image matching algorithms. The dataset to be used is as follows:

- 1) An image with repetitive patterns
- 2) An image with a cluttered background
- 3) An image of a planar object

These images are pictures taken from an Infinix hot note's rear mobile camera with a resolution of 8MP. Altogether the images correspond to indoor and outdoor conditions. The original images used were 3840x2160. However, these images were too large for the algorithms and hence were resized by a scale factor of 1/8 for efficient processing.

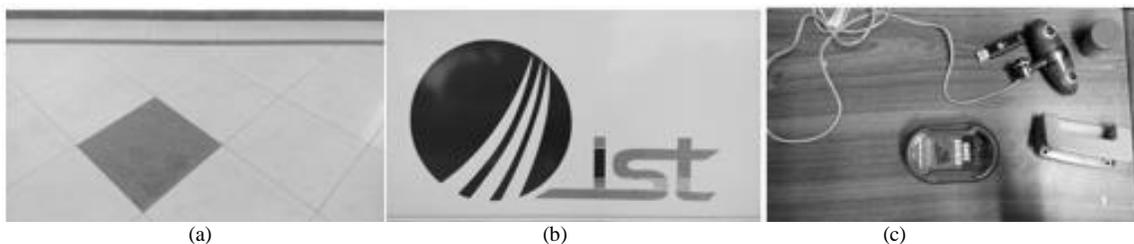


Fig. 1. Floor tiles at the Institute of Space technology (b) A logo of Institute of Space Technology (c) A shoe shiner in a cluttered scene

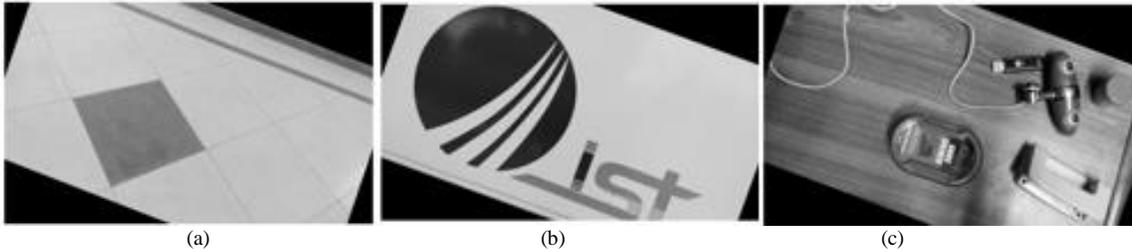


Fig. 2. (a) Floor tiles at the Institute of Space technology rotated at an angle of  $30^\circ$  (b) A logo of Institute of Space Technology rotated at an angle of  $30^\circ$  (c) A shoe shiner in a cluttered scene rotated at an angle of  $30^\circ$

Three of the most prevalent matching algorithms that are to be used for the study are SIFT, SURF and FAST.

SIFT is perhaps the most prevalent image matching algorithm. It can equate images under various scales, orientations and illuminations. However, it was found to be substantially slow. Various implementations for the code are available on the web. For this study the implementation being used was initially created by D. Alvaro and J.J. Guerrero, Universidad de Zaragoza and then modified by D. Lowe.

SURF, published after SIFT, was proposed to overcome the shortcomings of SIFT which include the computational cost and the execution time. The SURF implementation used in this study is a direct translation of the Open Surf C# code of Chris Evans.

FAST, first developed by Trajkovic and Hedley in 1998, is the lone feature based algorithm being used for the study. The implementation used was published by Edward Rosten (Rosten and Drummond, 2009). However, a matching module was not available for FAST as a MATLAB implementation. Though a bit unfair but a proportionality analysis could be performed to help keep it as fair as possible. All these implementation are tested using MATLAB R2015a on an Acer Aspire V5 with a 1.9GHz processor and 4.00GB RAM.

#### IV. EXPERIMENTATION

Each algorithm will be used to test the images shown above and these, consequently, will be tested in diverse categories with the final and chief goal of the evaluation of the intrinsic algorithm features that differentiate it from others. Parameters such as speed, number of feature and number of matches will be assessed. The images in this experiment will be taken in pairs with a rotation less than  $30^\circ$  from the image to its corresponding pair. A rotation of 30 degrees is chosen as it is the typical maximum value for most of the algorithms to perform a reliable research.

Once the match pairs are selected, the algorithms are applied to each image pair to quantify the number of features detected by each of the algorithm. However, the number of features detected is not a good measure of performance by itself because detecting 10 important features is better than detecting 10 features that have a lesser chance of finding a match pair. Therefore, a visual scrutiny was performed to manually match 10 features per image pair. Then these features were checked for a match by applying the algorithms to the images. The obtained results were then tested for the number of errors sustained. Type I and Type II error classification was

used for the process where Type I errors are the ones in which real matches are not sensed by the algorithms and Type II are the ones in which a feature is mismatched. The efficiency of the algorithm is calculated using the relation:

$$\%Efficiency = \frac{2 \times \text{No. of matches}}{\text{Total No. of features detected}}$$

The efficiency of each algorithm are presented in table II, III and IV. The results show that the number of matches are significantly lesser than the number of features detected. Therefore, it establishes that the number of matches are by themselves not very good. What is needed is to compare that how much of features that are detected by the algorithm are actually useful in the process of matching them to their correspondents.

#### V. RESULTS

The experimental follow up produced some likely results especially with the feature detection components of the algorithms. SURF detects better quality features than SIFT, in general, as SIFT fails to match a large number of features that it detects.

It is vital to remember that the analysis does not compare the types of images most suited to the algorithms. This can be observed in the following results as SIFT and SURF both being textured based algorithm present similar results on same images.

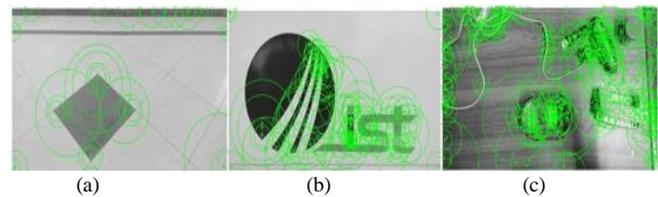


Fig. 3. SIFT feature detection on (a) Floor tiles at the Institute of Space technology (b) A logo of Institute of Space Technology (c) A shoe shiner in a cluttered scene

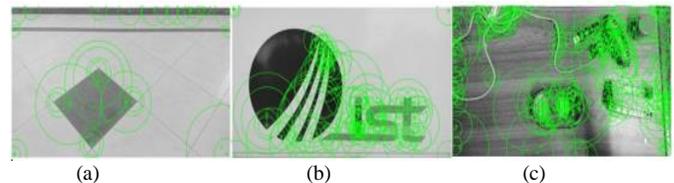


Fig. 4. SURF feature detection on (a) Floor tiles at the Institute of Space technology (b) A logo of Institute of Space Technology (c) A shoe shiner in a cluttered scene

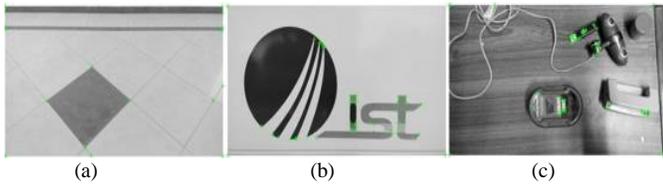


Fig. 5. FAST feature detection on (a) Floor tiles at the Institute of Space technology (b) A logo of Institute of Space Technology (c) A shoe shiner in a cluttered scene

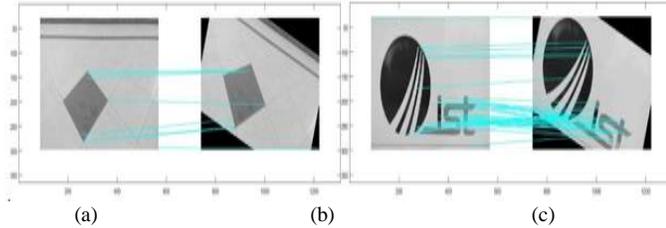


Fig. 6. SIFT matching on (a) Floor tiles at the Institute of Space technology (b) A logo of Institute of Space Technology

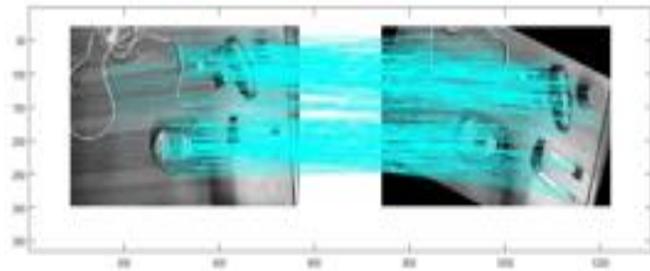


Fig. 7. SIFT matching on a shoe shiner in a cluttered scene



Fig. 8. SURF matching on (a) Floor tiles at the Institute of Space technology (b) A logo of Institute of Space Technology (c) A shoe shiner in a cluttered scene

TABLE I. FEATURES DETECTION USING SIFT, SURF AND FAST

Features Detected				
Pair	Image	SIFT	SURF	FAST
Floor	A	209	117	17
	B	129	93	10
IST logo	A	333	171	97
	B	162	83	112
Cluttered scene	A	675	552	47
	B	642	481	32

TABLE II. FEATURES MATCHING AND MATCHING EFFICIENCY USING SIFT, SURF AND FAST

	SIFT		
	Features	Matches	%Efficiency
Floor	209	19	11.24
	129		
IST logo	333	50	51.28
	162		
Cluttered scene	675	226	34.32
	642		
	SURF		
	Features	Matches	%Efficiency
Floor	117	29	27.62
	93		
IST logo	171	42	33.07
	83		
Cluttered scene	552	130	25.17
	481		
	FAST		
	Features	Matches	%Efficiency
Floor	17	9	58.06
	14		
IST logo	97	26	24.88
	112		
Cluttered scene	47	22	55.7
	32		

## VI. CONCLUSIONS

Given that SIFT is an established robust feature detector, it is not surprising that a large amount of features are detected in the images. SIFT detects a large amount of features in textured images as evident from the results. However, a lot of features it detected were in places with insufficient information for matching it later. But still the high number of features detected compensate for the features in areas with less information.

Following SIFT, SURF also detects a high number of features. But the difference between the features detected by each of the algorithm is significant. SURF also being a texture-based algorithm works best on textured images but doesn't give a lot of features with planar ones. Although SURF finds less features, it finds more features in areas with more information as compared to SIFT. This gives a positive sign about the algorithm as the features detected by the algorithm are more likely to be matched.

FAST, in contrast to its competitors, detects limited features. Being a feature based algorithm this could be expected of the algorithm. But such algorithms limit themselves to detect features with high contrast to the feature surroundings. Therefore, the features detected were among the most robust ones. FAST works best with planar images and the quickest too. This proves our hypothesis that the algorithm used depends on the type of image being used.

## VII. FUTURE CONTRIVES

Future prospects of this research include testing the accuracy of image matching algorithms in detecting a single object in a scene and an in-depth analysis of the algorithm structure to perform an automated assessment of the algorithm's performance in various scenarios.

### REFERENCES

- [1] Babbar, G., Punam Bajaj, Anu Chawla, and Monika Gogna. A comparative study of image matching algorithms. *International Journal of Information, Technology and Knowledge Management*. July-December. 2(2): 337-339. 2010.
- [2] Bay, H., A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision ECCV 2006*, Vol. 3951. *Lecture Notes in Computer Science*. p. 404-417. 2006
- [3] Dorkó, G., and C. Schmid. Selection of scale-Invariant parts for object class recognition. *Proceedings on Ninth IEEE International Conference on Computer Vision 2003*, 13-16 Oct. 2003, Nice, France. 2003.
- [4] Harris, C. A combined corner and edge detector. *Fourth Alvey Vision Conference*, Manchester, UK. p. 147-151. 1998.
- [5] Juan, L., and O. Gwun. A comparison of SIFT, PCA-SIFT, and SURF. *International Journal of Image Processing (IJIP)* 3(4): 143-152. 2009.
- [6] Ke, Y., and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)* – 2: 506-513. 2004.
- [7] Lowe, D.G. Object recognition from local scale-invariant features. *The proceeding of the seventh IEEE International Conference on Computer Vision 1999*. 2: 1150-1157. 1999.
- [8] Lowe, D.G. Distintive image feature from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2): 91-110. 2004.
- [9] Mikolajczyk, K, and. C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision* 60(1): 63-86. 2004.
- [10] Trajkovic, M. and M. Hedley. FAST corner detector. *Image and Vision Computing* 16: 75-87. 1998.
- [11] Guerrero, Maridalia. A Comparative Study of Three Image Matching Algorithms: Sift, Surf, and Fast. *All Graduate Theses and Dissertations*. Paper 1040. 2011.
- [12] Rosten, E., Drummond, T. 2005. Fusing points and lines for high performance tracking. In *10th IEEE International Conference on Computer Vision*. Volume 2., Beijing, China, Springer. 1508-1515
- [13] Rosten, E., and T. Drummond. Machine learning for high-speed corner detection, *IEEE Trans. Pattern Analysis and Machine Intelligence* 2006. p. 430-443. 2006.
- [14] Lindeberg, T. Scale invariant feature transform. *Scholarpedia* 7(5), 10491. 2012
- [15] B. Pinto and P. R. Anurenjan, "Video stabilization using Speeded Up Robust Features," *Communications and Signal Processing (ICCSPP)*, 2011 International Conference on, Calicut, pp. 527-531.
- [16] Elmar Mair, Gregory D. Hager, Darius Burschka, Michael Suppa and Gerhard Hirzinger. Adaptive and Generic Corner Detection Based on the Accelerated Segment Test. *Computer Vision ECCV 2010*, Vol. 6312. *Lecture Notes in Computer Science*. p. 183-196. 2010.