



Cairo University
Egyptian Informatics Journal

www.elsevier.com/locate/eij
www.sciencedirect.com



ORIGINAL ARTICLE

A comparative study of image low level feature extraction algorithms

M.M. El-gayar ^{*}, H. Soliman, N. meky

Information Technology Department, Faculty of Computers and Information System, Mansoura University, Egypt

Received 26 November 2012; revised 12 June 2013; accepted 17 June 2013

Available online 2 August 2013

KEYWORDS

SIFT;
PCA-SIFT;
F-SIFT;
SURF;
FAST

Abstract Feature extraction and matching is at the base of many computer vision problems, such as object recognition or structure from motion. Current methods for assessing the performance of popular image matching algorithms are presented and rely on costly descriptors for detection and matching. Specifically, the method assesses the type of images under which each of the algorithms reviewed herein perform to its maximum or highest efficiency. The efficiency is measured in terms of the number of matches found by the algorithm and the number of type I and type II errors encountered when the algorithm is tested against a specific pair of images. Current comparative studies assess the performance of the algorithms based on the results obtained in different criteria such as speed, sensitivity, occlusion, and others. This study addresses the limitations of the existing comparative tools and delivers a generalized criterion to determine beforehand the level of efficiency expected from a matching algorithm given the type of images evaluated. The algorithms and the respective images used within this work are divided into two groups: feature-based and texture-based. And from this broad classification only three of the most widely used algorithms are assessed: color histogram, FAST (Features from Accelerated Segment Test), SIFT (Scale Invariant Feature Transform), PCA-SIFT (Principal Component Analysis-SIFT), F-SIFT (fast-SIFT) and SURF (speeded up robust features). The performance of the Fast-SIFT (F-SIFT) feature detection methods are compared for scale changes, rotation, blur, illumination changes and affine transformations. All the experiments use repeatability measurement and the number of correct matches for the evaluation measurements. SIFT presents its stability in most situations although its slow. F-SIFT is the fastest one with good performance as the same as SURF, SIFT, PCA-SIFT show its advantages in rotation and illumination changes.

© 2013 Production and hosting by Elsevier B.V. on behalf of Faculty of Computers and Information, Cairo University.

^{*} Corresponding author. Tel.: +20 1061936935.

E-mail addresses: algyar_1990@yahoo.com (M.M. El-gayar), hsoliman@mans.edu.eg (H. Soliman), nona762005@hotmail.com (N. meky).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.



Production and hosting by Elsevier

1. Introduction

Feature detection and image matching represent two important tasks in computer vision, computer graphics, photogrammetric and all images' applications. Their application continues to grow in a variety of fields day by day. From simple photogrammetric tasks such as feature recognition, to the development of sophisticated 3D modeling software and image's search engine, there are several applications where image matching algorithms play an important role. Moreover, this has been a very active area of research in the recent decades and as indicated by the tremendous amount of work and documentation published around this. More than a decade ago, the applications associated with 2D and 3D models and object reconstruction were mainly for the purpose of visual inspection and robotics. Today, these applications now include the use of 2D and 3D models in computer graphics, virtual reality, communication and others. But achieving highly reliable matching results from a pair of images is the task that some of the most popular matching methods are trying to accomplish. But none have been universally accepted.

It seems that the selection the adequate method to complete a matching task significantly depends on the type of image to be matched and in the variations within an image and its matching pair in one or many of the following parameters: (a) *Scale*: At least two elements of the set of images views have different scales. (b) *Occlusion*: Is the concept that two objects that are spatially separated in the 3D world might interfere with each other in the projected 2D image plane. (c) *Orientation*: The images views are rotated with respect to each other. (d) *Affine Transformation*: Whether is a planar, textured or edgy object. (e) *Blurring*: is the apparent streaking of rapidly moving objects in a still image or a sequence of images. (f) *Illumination*: Changes in illumination also represent a typical problem for accurate feature matching [1,2,3].

Comparative studies have been published and available assessing the performance of the image matching algorithms methods without other aspects like (time, cost and power consumption) but this study overcomes some of the limitations of the current comparative studies by incorporating the analysis of the algorithms using different scenes to determine under which circumstances they will provide optimum results. In [4], they showed how to compute the repeatability measurement of affine region detectors also in [5] the image was characterized by a set of scale invariant points for indexing.

2. Related work

During the process of searching for documentation on 2D modeling, a lot of work was found that addresses the early feature detection and the posterior image matching. Most of the early implementations developed seemed to work well under certain limited image condition. The real challenge for those authors was to achieve true invariant feature detection under any image such (a) Consistency, detected positions should be insensitive to the noise, scale, orientation, cluttered, illumination. (b) Accuracy, should be detected as close as possible to the correct positions and features; (c) Speed, should be faster enough.

Some researches focused on the application of algorithms such as automatic image mosaic technique based on SIFT

[6,7], stitching application of SIFT [8–11] and Traffic sign recognition based on SIFT [10]. Ke and Sukthankar [12] gave some comparisons of SIFT and PCA-SIFT. PCA is well-suited to represent keypoint patches but observed to be sensitive to the registration error. In [13], the author used Fast-Hessian detector which is faster and better than Hessian detector. Section 3 will show more details of the three methods and their differences.

The first attempt towards digital image recognition was the color-based algorithm (color histogram or color distributive features). This practice although effective had many limitations. Color histogram was successful and faster in detecting color distribution features in any given images meeting basic requirements. But it was unsuccessful in matching large set of images and no satisfies the following criteria (Consistency, Accuracy) [14].

The second attempts towards digital image recognition were limited to the identification of corners and edges. The beginnings of feature detection can be tracked with the work of Harris and Stephen and the later called *Harris Corner Detector*. Harris was successful in detecting robust features in any given image. But since it was only detecting corners, his work suffered from a lack of connectivity of feature-points which represented a major limitation for obtaining major level descriptors such as surfaces and objects. Almost a decade after the Harris Detector was published; a new corner detector algorithm called FAST (*Features from Accelerated Segment Test*) was presented.

The third attempt towards digital image recognition was limited to achieve reliable image matching from textured image with cluttered backgrounds. Before this, it is important to know that feature-based algorithms have been widely used as feature point detectors because colors, corners and edges correspond to image colors and locations respectively with high information content, meaning this that they can be matched between images. But the feature-based detectors only perform accurately when the objects to be matched have a same color or a distinguishable corner or edge. Furthermore, the feature-based algorithms do not perform as good as expected when images are subjected to variations in color's distribution, scale, illumination, rotation or affine transform.

To overcome these limitations, a new class of image matching algorithm was developed simultaneously. These algorithms are known as texture-based algorithms because of their capability to match features between different images despite of the presence of textured backgrounds and lack of planar and well-defined edges. One of the first attempts towards this novel approach was undertaken by David Lowe.

Lowe [15] presented SIFT (*Scale Invariant Feature Transform*) for extracting distinctive invariant features from images that can be invariant to image scale and rotation [10,15,16]. Then it was widely used in image mosaic, recognition, retrieval and etc. After Lowe, Ke and Sukthankar used PCA (*Principal Component Analysis-SIFT*) to normalize gradient patch instead of histograms [12]. They showed that PCA-based local descriptors were also distinctive and robust to image deformations. But the methods of extracting robust features were still very slow. Bay et al. SURF (*speeded up robust features*) and used integral images for image convolutions and Fast-Hessian detector [13]. Their experiments turned out that it was faster and it works well.

3. Overview of image low level feature extraction algorithms

3.1. Feature-based algorithm

3.1.1. Color histogram (color detector)

A color histogram is a representation of the distribution of colors in an image. For digital images, a color histogram represents the number of pixels that have colors in each of a fixed list of color ranges that span the image's color space, the set of all possible colors. Color histograms are flexible constructs that can be built from images in various color spaces, whether RGB, or any other color space of any dimension [14,17].

The main drawback of histograms for classification is that the representation is dependent of the color of the object being studied, ignoring its shape and texture. Color histograms can potentially be identical for two images with different object content which happens to share color information. Conversely, without spatial or shape information, similar objects of different color may be indistinguishable based solely on color histogram comparisons. There is no way to distinguish a red and white cup from a red and white plate. Put another way, histogram-based algorithms have no concept of a generic 'cup', and a model of a red and white cup is no use when given an otherwise identical blue and white cup.

3.1.2. FAST (corner detector)

FAST (*Features from Accelerated Segment Test*) algorithm based on the SUSAN (Smallest Univalve Segment Assimilating Nucleus) corner criterion [18]. For feature detection, SUSAN places a circular mask over the pixel to be tested (the nucleus). The region of the mask is M and a pixel in this mask is represented by m and every pixel is compared to the nucleus using the comparison function:

$$C(m) = e - \left(\frac{(I(m) - I(m0))}{t} \right)^6 \quad (1)$$

where t determines the radius, and the power of the exponent has been determined empirically.

The area of the SUSAN is given by:

$$n(M) = \sum_{m \in M} C(m) \quad (2)$$

With FAST, the detection of corners was prioritized over edges as they claimed that corners are one of the most intuitive types of features that show a strong two dimensional intensity change, and are therefore well distinguished from the neighboring points. According to a comparative study of the existing corner detectors based on the above criteria (Consistency, Accuracy and speed) was found that most of these detectors satisfied one of the criterions but failed in the others [18].

3.2. Texture-based algorithm

3.2.1. SIFT detector

SIFT (*the Scale Invariant Feature Transform*) consists of four major stages: (a) scale-space detection, (b) keypoint localization, (c) orientation assignment and (d) keypoint descriptor. The first stage used difference-of-Gaussian (DOG) function to identify potential interest points [15], which were invariant to scale and orientation. DOG was used instead of Gaussian to improve the computation speed [15,16,19].

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ = L(x, y, k\sigma) - L(x, y, \sigma) \quad (3)$$

where $*$ is the convolution operator, $G(x, y, \sigma)$ is a variable-scale Gaussian, $I(x, y)$ is the input image $D(x, y, \sigma)$ is Difference of Gaussians with scale k times. In the keypoint localization step, they are rejected the low contrast points and eliminated the edge response. Hessian matrix was used to compute the principal curvatures and eliminate the keypoints that have a ratio between the principal curvatures greater than the ratio. An orientation histogram was formed from the gradient orientations of sample points within a region around the keypoint in order to get an orientation assignment [15]. According to experiments, the best results were achieved with a 4×4 array of histograms with 8 orientation bins in each. So the descriptor of SIFT that was used is $4 \times 4 \times 8 = 128$ dimensions [7,10].

3.2.2. PCA-SIFT detector

PCA-SIFT (*Principal Component Analysis-SIFT*) is a standard technique and new algorithm emerged as an attempt to improve SIFT, for dimensionality reduction and eliminate the computa-

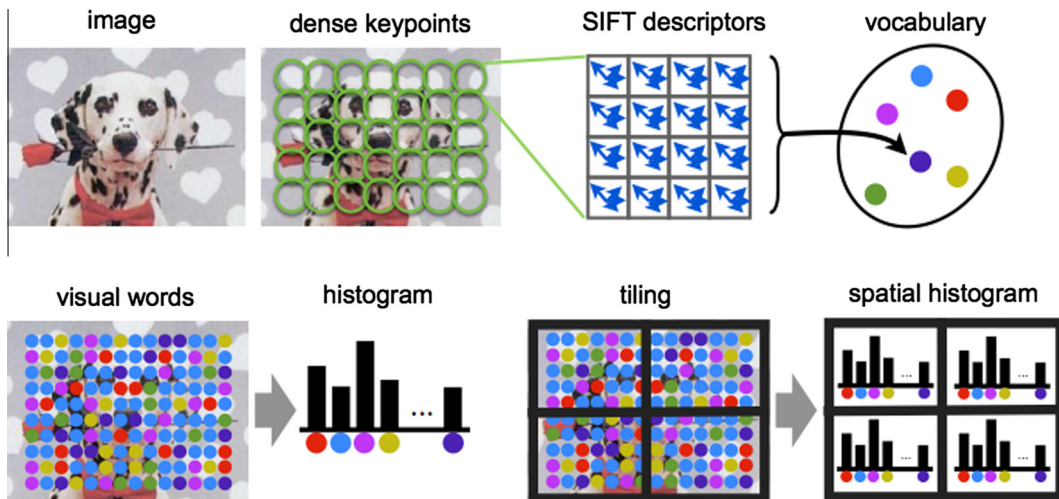


Figure 1 Fast-SIFT steps.

tional costs carried with Lowe's implementations. [12], which is well-suited to represent the keypoint patches and enables us to linearly-project high-dimensional samples into a low-dimensional feature space. In other words, PCA-SIFT uses PCA instead of smoothed weighted histograms to normalize gradient patch [12]. The feature vector is significantly smaller than the standard SIFT feature vector, and it can be used with the same matching algorithms. PCA-SIFT, like SIFT, also used Euclidean distance to determine whether the two vectors correspond to the same keypoint in different images. In PCA-SIFT, the input vector is created by concatenation of the horizontal and vertical gradient maps for the 41×41 patch centered to the keypoint, which has $2 \times 39 \times 39 = 3042$ elements [12].

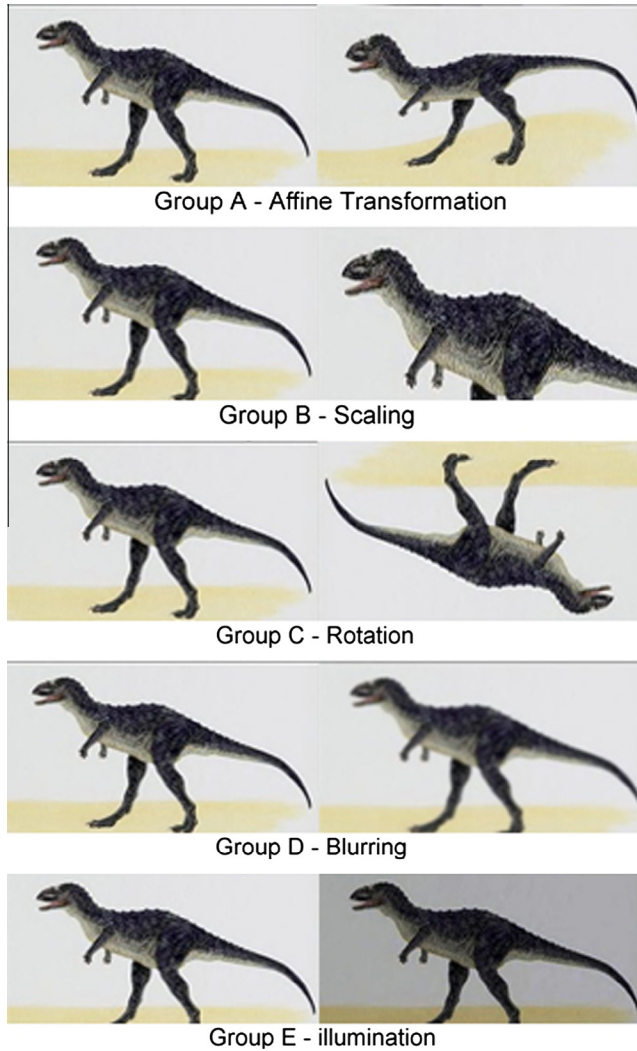


Figure 2 Sample images in test.

According to PCA-SIFT, fewer components requires less storage and will be resulting to a faster matching. The PCA-SIFT achieved the ability to speed up the SIFT's matching process by an order of magnitude, but it was proved to be less distinctive than SIFT.

3.2.3. SURF detector

The Speed-Up Robust Feature detector (SURF) was conceived to ensure high speed in three of the feature detection steps: detection, description and matching. Unlike PCA-SIFT, SURF speeded up the SIFT's detection process without scaring the quality of the detected points.

SIFT and SURF algorithms employ slightly different ways of detecting features. SIFT builds an image pyramids, filtering each layer with Gaussians of increasing sigma values and taking the difference. On the other hand, SURF creates a "stack" without 2:1 down sampling for higher levels in the pyramid resulting in images of the same resolution [6].

In keypoint matching step, the nearest neighbor is defined as the keypoint with minimum Euclidean distance for the invariant descriptor vector. Lowe used a more effective measurement that obtained by comparing the distance of the closest neighbor to that second-closest neighbor [15].

3.2.4. F-SIFT detector

F-SIFT (Fast-SIFT) consists of the same four major stages of SIFT: (a) scale-space detection, (b) keypoint localization, (c) orientation assignment and (d) keypoint descriptor and feature vector quantized into visual words and the feature vector is significantly smaller than the standard SIFT feature vector. The frequency of each visual word is then recorded in a histogram for each tile of a spatial tiling as shown in Fig. 1. The final feature vector for the image is a concatenation of these histograms.

4. Experiments and results

4.1. Evaluation measurement

The evaluation measurement is RANSAC (Random Sample Consensus), which is used to reject inconsistent matches. The inlier is a point that has a correct match at the input image. Our goal is to obtain the inliers and reject outliers in the same time [5]. As Eq. (4), the probability that the algorithm never selects a set of m points which all are inliers is $1 - p$:

Table 2 Affine transformation comparison on time. Using group A of Fig. 1, data represents the total number of matches for each method on the time.

	FAST	F-SIFT	PCA-SIFT	SIFT	SURF
Total matches	81	111	50	175	298
Total time (ms)	99.4	38.4	30.5	52.8	59

Table 1 Feature detection (descriptors) comparison on time on affine transformation problem. Using group A of Fig. 2, total time is the time of finding all descriptors.

	FAST		F-SIFT		PCA-SIFT		SIFT		SURF	
Total descriptors (im1, img2)	81	40	289	346	50	33	1600	1600	298	241
Total time (ms)	924	644	209	215	30.5	20	1132	1126	113	95

Table 3 Feature detection (descriptors) comparison on time on Scale changes problem. Using group B of Fig. 2, total time is the time of finding all descriptors.

	FAST		F-SIFT		PCA-SIFT		SIFT		SURF	
Total descriptors (im1, img2)	81	78	299	352	50	40	1560	1565	298	298
Total time (ms)	896	658	209	225	30.5	29	1163	1165	111	127

Table 4 Scale changes comparison on time. Using group B of Fig. 1, data represents the total number of matches for each method on the time.

	FAST	F-SIFT	PCA-SIFT	SIFT	SURF
Total matches	81	99	20	136	298
Total time (ms)	141	17	15.5	543.4	6

$$1 - p = (1 - w^m)^k \quad (4)$$

where m is the least number of points that needed for estimating a model, k is the number of samples required and w is the probability that the RANSAC algorithm selects inliers from the input data. The RANSAC repeatedly guess a set of mode of correspondences that are drawn randomly from the input set. We can think the inliers as the correct match numbers. In the following experiments, matches mean inliers [19].

In this paper, the methods that were used are all based on OPENCV and VL-feat. We use the standard image dataset, which includes the general deformations, such as scale changes, view changes, illumination changes and rotation. As shown in Fig. 2. All the experiments work on LAPTOP processor 1.8 GHz and 1.0 GB RAM, with Windows 7 as an operating system. Time evaluation is a relative result, which only shows the tendency of the methods' time cost. There are factors that influenced on the results such as the size (512 * 512) and quality of the image, image types (JPG) and the parameters of the algorithm (threshold = 1.5).

4.2. Affine transformation on processing time

The first part of the experiment shows the performance of affine-transformation invariant. Table 1 shows that SURF is the fastest one, SIFT is the slowest but it finds most descriptors and F-SIFT faster than sift and find more descriptor than SURF. Table 2 shows that F-SIFT is the fastest one, SURF is the slowest but it finds most matches.

4.3. Scale changes on processing time

The second experiment shows the performance of scale invariant. Table 3 shows that SURF is the fastest one, SIFT is the slowest but it finds most descriptors and F-SIFT faster than sift and find more descriptor than SURF. Table 4 shows that SURF better than F-SIFT over matches and time.

Table 6 Rotation changes comparison on time. Using group C of Fig. 1, data represents the total number of matches for each method on the time.

	FAST	F-SIFT	PCA-SIFT	SIFT	SURF
Total matches	79	237	200	662	298
Total time (ms)	155	15	17	572.4	20

4.4. Rotation changes on processing time

The third experiment shows the manipulated of rotation on the methods. As shown in group C of Fig. 1, the image rotates 180 degrees. Table 5 shows that SURF is the fastest one, SIFT is the slowest but it finds most descriptors and F-SIFT faster than sift and find more descriptor than SURF like experiments above. Table 6 shows that F-SIFT better than SURF over time and matches are almost equal.

4.5. Blurring changes on processing time

This fourth experiment uses Gaussian blur like the images in group D of Fig. 2. The radius of the blur changes to 1.5. Table 7 shows that SURF is the fastest one, SIFT is the slowest but it finds most descriptors and F-SIFT faster than sift and find more descriptor than SURF like experiments above. Table 8 shows that PCA-SIFT the fastest and F-SIFT better than SURF over time and matches are almost equal.

4.6. Illumination changes on processing time

This fifth experiment shows the illumination effects of the methods. As shown in group G of Fig. 2, from data 1 to data 6, the brightness of the image gets lower and lower. Table 9 shows that SURF is the fastest one, SIFT is the slowest but it finds most descriptors and F-SIFT faster than sift and find more descriptor than SURF like experiments above. Table 10 shows that SURF the fastest and SURF better than F-SIFT over time and matches. Table 10 shows the repeatability of illumination changes. SURF has the largest repeatability, F-SIFT shows as good performance as SURF, which coincides with [12,13].

4.7. Discussion

Table 11 and Fig. 3 show the results of all experiments. It also shows that there is no best method for all deformation. Hence,

Table 5 Feature detection (descriptors) comparison on time on rotation changes problem. Using group C of Fig. 2, total time is the time of finding all descriptors.

	FAST		F-SIFT		PCA-SIFT		SIFT		SURF	
Total descriptors (im1, img2)	81	79	299	310	50	39	1552	1552	298	294
Total time (ms)	531	516	203	204	30.5	29.3	1125	1127	110	109

Table 7 Feature detection (descriptors) comparison on time on blurring changes problem. Using group D of Fig. 2, total time is the time of finding all descriptors.

	FAST		F-SIFT		PCA-SIFT		SIFT		SURF	
Total descriptors (im1, img2)	81	6	299	350	50	32	1500	1250	298	203
Total time (ms)	516	518	203	220	30.5	27.5	1127	1125	111	97

Table 8 Blurring changes comparison on time. Using group D of Fig. 1, data represents the total number of matches for each method on the time.

	FAST	F-SIFT	PCA-SIFT	SIFT	SURF
Total matches	6	227	80	688	296
Total time (ms)	75	21	12.5	584	40

when choosing a feature detection method, make sure which most concerned performance is. The result of this experiment is not constant for all cases. Changes of an algorithm can get a new result, find the nearest neighbor instead of KNN or use an improved RANSAC.

Let's discuss the performance of each algorithm for a given type of images. The results of this are presented in Fig. 3:

- SIFT detected the most feature points on image. This was expected because of the large amount of texture of this image (visually appreciated). On the other hand, SIFT had the lowest detection with image 5 corresponding to the first view of the Haddock Boat. This also confirm the hypothesis of SIFT performing at the highest levels when tested on textured images. This can be also checked by looking at the results for the other images.
- SURF is also a textured based matching algorithm but it seemed to get confused in textured images with illumination changes. Because of this, it did not have its best performance with this pair. The image with most features detected by SURF was the corresponding second one for SIFT because of less feature detection occurred. This proves that SIFTS and SURF being texture-based algorithms, do not perform well when tested on planar images.
- F-SIFT, conversely to SIFT and SURF, detected more features in the pair comprised by the Haddock boat. Although in general the amount of features detected by F-SIFT are significantly less than SIFT or SURF, it is appreciated that the best performance of this type of algorithms (feature-based) will be enhanced if the appropriate set of image is selected. This is an unexpected result from F-SIFT due to the large number of

Table 10 Illumination changes comparison on time. Using group E of Fig. 1, data represents the total number of matches for each method on the time.

	FAST	F-SIFT	PCA-SIFT	SIFT	SURF
Total matches	25	197	20	683	198
Total time (ms)	71	18	15.5	539	4

corners and edges within this image, but this is presumably because of the low contrast and poor illumination at the corners and edges of the brick.

FAST appears significantly good in time and bad in other things. SIFT's matching success attributes to that its feature representation has been carefully designed to be robust to localization error. As discusses PCA is known to be sensitive to registration error. Using a small number of dimensions provides significant benefits in storage space and matching speed [11]. F-SIFT appear good and stable like SURF because of using KD-Tree to represent and index the descriptors. SURF shows its stability and fast speed in the experiments. It is known that 'Fast-Hessian' detector that used in SURF is more than 3 times faster that DOG (which was used in SIFT) and 5 times faster than Hessian-Laplace.

5. Conclusion and future work

This paper has evaluated five feature detection methods for image deformation. SIFT is slow and not good at scaling changes, while it is invariant to rotation, illuminate changes and affine transformations. Fast SIFT is faster than normal SIFT and appear good in different aspects but SIFT is better performance than fast SIFT. SURF is fast and has good performance as the same as SIFT, but it is not stable to rotation and illumination changes.

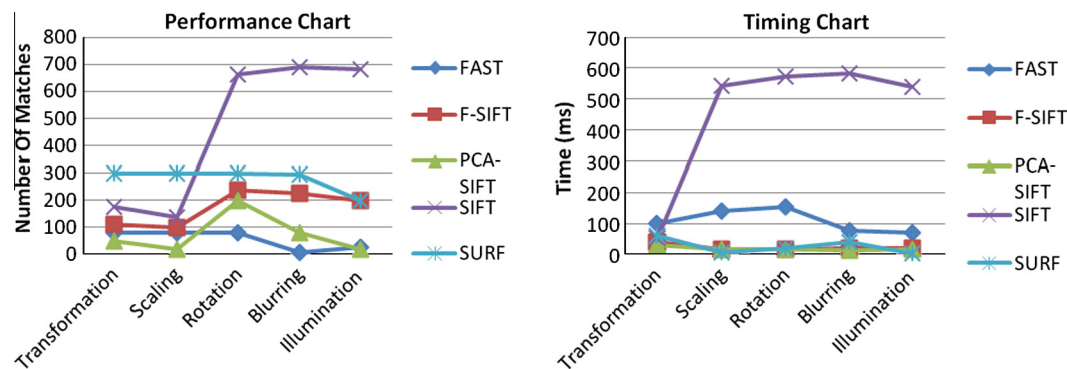
It was concluded that F-SIFT has the best overall performance above SIFT and SURF but it suffers from detecting very few features and therefore matches. It is recommended that the properties of this algorithm to be improved by creating a new implementation provided with a matching component. It is necessary to improve F-SIFT by increasing the amount of features it can detect. But special care should be taken to preserve the robustness of the algorithm and avoid the detection of useless features. Future research in this area should focus on testing

Table 9 Feature detection (descriptors) comparison on time on illumination changes problem. Using group E of Fig. 2, total time is the time of finding all descriptors.

	FAST		F-SIFT		PCA-SIFT		SIFT		SURF	
Total descriptors (im1, img2)	81	25	299	306	50	30	1552	1552	298	181
Total time (ms)	541	523	204	199	30.5	25	1125	1123	114	91

Table 11 Conclusion of all the experiments.

Methods	Timing	Transformation	Scaling	Rotation	Blurring	Illumination
FAST	Common	Common	Common	Bad	Bad	Bad
SIFT	Bad	Good	Good	Best	Best	Best
PCA-SIFT	Good	Bad	Bad	Common	Common	Bad
F-SIFT	Best	Good	Best	Good	Good	Good
SURF	Good	Best	Best	Good	Good	Good

**Figure 3** Performance and time charts as a conclusion of all the experiments.

the accuracy of the algorithms in detecting a single object within a scene. Many of the new arising needs in photogrammetric address this problem and this work could be a first step towards a more in depth study. And, also improve semantic techniques with F-SIFT extraction feature algorithm to remove semantic gap between high-level semantic perception of human and low-level features of an image.

References

- [1] Mikolajczyk K, Schmid C. A performance evaluation of local descriptors. *IEEE Trans Pattern Anal Mach Intel* 2005;27(10):1615–30.
- [2] Heo Y, Lee K, Lee S. Illumination and camera invariant stereo matching. *Comput Vision Pattern Recognit CVPR* 2008:1–8.
- [3] Sarin S, Fahrmaier M, Wanger M, Kameyama W. Holistic feature extraction for automatic image annotation. In: *Fifth FTRA international conference on multimedia and ubiquitous engineering*; 2011.
- [4] Mikolajczyk K, Tuytelaars T, Schmid C, Zisserman A, Matas J, Schaffalitzky F, Kadir T, Gool LV. A comparison of affine region detectors. *IJCV* 2006;65(1/2):43–72.
- [5] Mikolajczyk K, Schmid C. Indexing based on scale invariant interest points. In: *Proceedings of eighth international conference of computer vision*; 2001, p. 525–31.
- [6] Yang zhan-long, Guo bao-long. Image mosaic based on SIFT. In: *International conference on intelligent information hiding and multimedia, Signal Processing*; 2009. p 1422–5.
- [7] Salgian A. Object recognition using local descriptors. In: *International symposium on visual computing, Lake Tahoe, NV*; 2007. p. 709–17.
- [8] Brown M, Lowe D. Recognizing panoramas. In: *Proceedings of ninth international conference computer vision*; 2003, p. 1218–27.
- [9] Cheng-Yuan Tang, Yi-Leh Wu, Maw-Kae Hor, Wen-Hung Wang. Sift descriptor for image matching under interference. In: *Machine learning and cybernetics international conference*. Vol. 6; 2008. p. 3294–300.
- [10] Heo Y, Lee K, Lee S. Illumination and camera invariant stereo matching. *CVPR* 2009:1–8.
- [11] Feng Gao, Shaohua Wei, Xuetong Wang. Image matching algorithm based on SIFT feature point. *Mod Electron Technol* 2010;18.
- [12] Ke Y, Sukthankar R. PCA-SIFT: A more distinctive representation for local image descriptors. In: *Proceedings of conference computer vision and pattern recognition*; 2004, p. 511–7.
- [13] Bay H, Tuytelaars T, Van Gool L. SURF: speeded up robust features. In: *9th European conference on computer vision*; 2006.
- [14] Jinxia L, Yuehong Q. Application of SIFT feature extraction algorithm on the image registration. In: *Tenth international conference on electronic measurement & instruments IEEE*; 2011.
- [15] Lowe D. Distinctive image features from scale-invariant keypoints. *IJCV* 2004;60(2):91–110.
- [16] Wang X. Robust image retrieval based on color histogram of local feature regions. Springer Netherlands; 2009.
- [17] Stokman H, Gevers T. Selection and fusion of color models for image feature detection. *Pattern Anal Mach Intel IEEE Trans* March 2008;29:371–81.
- [18] Rosten E, Porter R, Drummond T. FASTER and better: a machine learning approach to corner detection. *IEEE Trans Pattern Anal Mach Intel* 2010;32:105–19.
- [19] Takacs G, Chandrasekhar V, Chen D, Tsai S, Grzeszczuk R, Girod B. Unified real-time tracking and recognition with rotation invariant fast features. In: *IEEE Conference on computer vision and pattern recognition (CVPR)*; June 2010.