Local Detectors and Descriptors for Object Class Recognition

Faten A. Khalifa and Noura A. Semary
Faculty of Computers and Information, Menofia University, Menofia, Egypt
E-mail: {faten_a_khalifa, noura.samri}@ci.menofia.edu.eg

Hatem M. El-Sayed and Mohiy M. Hadhoud
Faculty of Computers and Information, Menofia University, Menofia, Egypt
E-mail: {hatem6803, mmhadhoud}@yahoo.com

Abstract—Local feature detection and description are widely used for object recognition such as augmented reality applications. There have been a number of evaluations and comparisons between feature detectors and descriptors and between their different implementations. Those evaluations are carried out on random sets of image structures. However, feature detectors and descriptors respond differently depending on the image structure. In this paper, we evaluate the overall performance of the most efficient detectors and descriptors in terms of speed and efficiency. The evaluation is carried out on a set of images of different object classes and structures with different geometric and photometric deformations. This evaluation would be useful for detecting the most suitable detector and descriptor for a particular object recognition application. Moreover, multi-object applications such as digilog books could change the detector and descriptor used based on the current object. From the results, it has been observed that some detectors perform better with certain object classes. Differences in performance of the descriptors vary with different image structures.

Index Terms—Local feature detectors, Local feature descriptors, Binary descriptors, RANSAC, Object recognition, Augmented reality, Digilog book.

I. INTRODUCTION

The field of computer vision has many various applications that basically depend on local feature detection and description. Examples of these applications are object recognition [1], object tracking [2], 3D reconstruction [3] image stitching [4] and visual mapping [5].

A feature detector selects points of interest that have unique content within an image. The key to feature detection is to find features that remain locally invariant so that you can detect them even in the presence of any type of deformations such as rotation, scale, illumination changes. An ideal feature detector should achieve repeatability, distinctiveness, locality, quantity, accuracy, and efficiency [6]. A feature descriptor (extractor) describes the region around each detected feature. Feature descriptors rely on image processing to transform a local pixel neighborhood into a compact vector representation. The resulting feature descriptors are then used for comparison and matching between neighborhoods regardless of any deformations. There are two types of descriptors; patch descriptors and binary descriptors, as will be discussed later. An ideal feature descriptor should be distinct, efficient and invariant to common deformations such as scale, rotation, image noise, and illumination changes. Moreover, the detector and the descriptor should be fast enough in order to be exploited in real-time applications.

The SIFT (Scale Invariant Feature Transform) [7] algorithm is considered the most important feature detector and descriptor and has been widely used in many applications. During the past decade, a lot of algorithms have been proposed with the aim of increasing the speed and efficiency of the feature detection and description process [8-12]. Many comparisons have been established between them [13-16] and between their different implementations [17, 18]. All these comparisons are carried out on a set of images with random structures and the results are reported for every deformation case. In addition, there is no best method for all deformations [15].

An object class recognition application could be faced with all types of deformations. Moreover, some object class recognition applications, such as multi-object tracking applications, deal with different object classes. For example, in a digilog book, many objects of different classes have to be detected and tracked.

The objective of this paper is to investigate the overall performance of the most recent feature detectors and descriptors on a set of images of different classes with all possible deformations. We want to know how detectors and descriptors deal with different image structures.

Our results would be useful by adding more flexibility and hence efficiency to object recognition applications. For single-object recognition, the most suitable (efficient) detector and descriptor could be determined beforehand based on the targeted object class. On the other hand, multi-object applications could be more flexible and efficient. Flexibility could be achieved by detecting the most suitable algorithm for each tracked object class.

Once the current object class is changed during the recognition process, the algorithm used could be automatically replaced by the most suitable one for that class.

The detectors included in the evaluation are SIFT, SUIRF (Speeded-Up Robust Features), ORB (Oriented FAST and Rotated BRIEF), FAST (Features from Accelerated Segment Test) and BRISK (Binary Robust Invariant Scalable Keypoints). The descriptors are SIFT, SURF, ORB, BRISK and BRIEF (Binary Robust Independent Elementary Features).

The organization of the remaining content is as follows: An overview of the algorithms and the related work are presented in section II. Section III presents the evaluation criteria for our evaluation and the data sets used as well as the implementation details. In section IV, the experimental results are discussed. Finally, the conclusion is reported in section V.

II. OVERVIEW AND RELATED WORK

A. Local feaure detectors overview

Despite its extreme slowness, the SIFT [7] algorithm has attracted a great attention due to its high efficiency. It detects features using Difference of Gaussians (DoG) which is an approximation of the little costly Laplacian of Gaussian (LoG). It builds a scale-space pyramid of filtered images. Such pyramid consists of a number of layers (octaves) and scales. Each octave is created by half-sampling the original image. Each scale is the result of convolving the corresponding octave with the Gaussian kernel using incremental scaling parameter. Then, adjacent layers are subtracted to produce the DoG pyramid. Features are identified as local maxima of the DoG at each scale. A Local maxima is obtained by comparing each pixel in the pyramid with its 8 neighbors as well as 9 pixels in the next scale and 9 pixels in the previous scales. DoG is sensitive to edges, so the Hessian matrix is used for eliminating this sensitivity. An orientation is assigned to each keypoint to achieve rotation invariance. Gradient magnitudes and directions are calculated for the neighboring pixels. A histogram with 36 bins covering 360 degrees is created. An orientation is obtained from peaks in the histogram.

SURF [8] is a powerful scale and rotation-invariant interest point detector which is faster than SIFT. It approximates LoG with Box Filters for finding scale-space. They are efficiently evaluated very fast using integral images, independently of their size. SURF is based on the determinant of the Hessian matrix which is efficiently calculated using Box filters. A scale-space representation is built from a number of octaves. An octave is a set of images filtered by a double scaled up kernel. Features are detected as local maxima in a $3 \times 3 \times 3$ neighborhood in the scale-space. Each keypoint is assigned an orientation by calculating the Haar wavelet responses in horizontal and vertical direction within a circular neighborhood. Such neighborhood is of radius $6s$ around the interest point, where s is the scale at which the keypoint was detected. The wavelet response can be easily found out using integral images at any scale. They are adequately weighted with a Gaussian kernel. All responses within a sliding orientation window of angle 60 degrees are summed to calculate the dominant orientation.

An extremely fast detector is FAST [11]. It is a corner detection method which is based on examining the image intensity of a circle of 16 pixels centered at a candidate pixel. The pixel is a corner if there is a set of n contiguous pixels in the circle which are all brighter\ darker than intensity plus\minus a predefined threshold. n was chosen to be twelve. A high-speed test was proposed to exclude a large number of non-corners. The test examines only pixels at the four compass directions. At least three of them must achieve the previously mentioned condition in order to realize that the candidate pixel is a corner. Machine learning for a corner detector can be used. The FAST algorithm is run on a set of training images to build a vector of the detected feature points. The vector is subdivided into 3 subsets: darker, similar and brighter. A new Boolean variable is defined which is true if the point is a corner and false otherwise. The ID3 algorithm (decision tree classifier) is used to query each subset using the Boolean variable. The decision tree so created is used for fast detection in other images. Non-maximum Suppression is used to solve the problem of detecting multiple interest points in adjacent locations. The disadvantages of FAST are that it is not robust to high levels of noise and it is dependent on a threshold.

An efficient extension to FAST is the recently proposed ORB [9] detector. It addresses the weaknesses of FAST (rotation and scale invariance). It uses FAST to detect keypoints. Then, it applies the Harris corner detector for quality measure. For scale invariance, it uses scale pyramid to produce multiscale-features like SIFT. For rotation invariance, it computes the intensity weighted centroid moment of the patch with located keypoint at center. The orientation is obtained from the direction of the vector from this corner point to the centroid. To improve the rotation invariance, moments are computed. It is a fast and robust feature detector.

BRISK [12] also extends FAST by searching for maxima not only in the image plane, but also in a 3D scale-space. So, it is a scale-invariant detector. It first creates a scale space pyramid which consists of a number of octaves and intra-octaves. The octaves are formed by half-sampling the original image. Then, it computes FAST score across this scale space. It applies non-maximal suppression on the pixel level and computes sub-pixel maximum across patch. It computes continuous maximum across scales. Image coordinates from scale space feature point detection are re-interpolated.

B. Local feaure descriptors overview

Patch descriptors [7, 8] are based on local histogram of gradients (HOG). They have been successfully used in various applications. They are computationally expansive. Even though SURF speeds up the computation using integral images, this still isn't fast enough for some

applications.

On the other hand, a recent approach is to use binary descriptors [9, 10, 12] which are considered efficient alternatives to their floating-point competitors as they enable faster processing while requiring less memory. This fast processing is because only local intensity comparisons are needed to form a binary string and using the hamming distance as a distance measure between two binary strings. They are especially suited for real time applications and low power devices.

SIFT [7] descriptor is based on the gradient magnitude and orientation computed at the scale of the considered keypoint. A region of size 16×16 around the keypoint is taken. It is divided into 16 sub-regions of size 4×4. For each sub-region, 8-bin orientation histogram is created. Those 16 histograms are concatenated together to form a final descriptor as a 128 dimensional feature vector in length. It is invariant to rotation, scale and illumination changes.

SURF [8] descriptor is a distribution-based descriptor and is also based on Haar wavelet responses. A region around the keypoint (detected in scale s) of size $20s$ is selected and split up into 4×4 sub-regions. For each sub-region, horizontal and vertical wavelet responses are calculated and a 4-dimensional feature vector is formed. 16 4-dimensional vectors are combined together to build a descriptor with total 64 dimensions. So, it is faster in computation and matching than SIFT, nevertheless it is still not suitable for real-time applications.

BRIEF [10] descriptor was the first binary descriptor published and is the simplest one. It does not have a precise sampling pattern or an orientation compensation mechanism. It randomly selects pairs of pixels for building the descriptor. There are five methods proposed by the authors for selecting the point pairs. One of them is the Gaussian distribution which achieves the best results. The descriptor simply constructs a binary string from a set of comparisons between pixel pairs in an image patch, disregarding the neighboring pixels. This makes it sensitive to noise. This sensitivity can be reduced by prior smoothing of the image such as using the Gaussian filter. It can be 128, 256, or 512 dimensional bitstring.

ORB [9] descriptor is considered a rotation-invariant BRIEF. It steers BRIEF according to the orientation of keypoints. For a feature set of n binary tests, define a $2 \times n$ matrix which contains the coordinates of these features. Then using the patch orientation and its rotation matrix, a steered (rotated) version of the matrix can be constructed. ORB learns the optimal set of sampling pairs, whereas BRIEF uses randomly chosen sampling pairs. It uses machine learning techniques. This maximizes the descriptor's variance and minimizes the correlation under various orientation changes. An ORB descriptor has only 32 bytes.

BRISK [12] descriptor differs from BRIEF and ORB as it uses a hand-crafted sampling pattern, that is composed out of concentric rings. The pixel intensity values around each sampling point are smoothed using the Gaussian kernel. It consists of 512 binary bits.

Separate pairs of pixels are divided into two subsets, long-distance pairs and short-distance pairs. Long-distance pairs are used to determine feature orientation by computing the local gradient between them and summing these gradients. Short-distance pairs are used for the intensity comparisons that build the descriptor, as with all binary descriptors. The short pairs are rotated by the orientation computed earlier and comparisons are made between them.

C. Related Work

In [13], Mikolajczyk et al. presented a comparison between affine covariant region detectors and concluded that combining several detectors would lead to the best performance. They also found that all the detectors have similar performance with all types of images and transformations. However, in their comparison between SIFT, PCA-SIFT (Principal Component Analysis) and SURF detectors, Juan and Gwun [15] found that SIFT was the most stable one in all the experiments except for time whereas SURF was the fastest with good results.

A comparison of local descriptors was introduced in [14]. It showed that the performance of the descriptor doesn't depend on the detector. Moreover, the SIFT descriptor was the best one for different image transformations except light changes. A comprehensive comparison between the recently proposed and state-of-the-art detectors and descriptors was introduced in [16]. It was in terms of number of features extracted, run time, speedup and repeatability for detectors and precision and recall for descriptors. Its conclusion was that binary descriptors are the best choice for time-constrained applications with good matching accuracy. In their work, Lankinen et al. [19] compared between a number of detectors and descriptors on different image classes. They used pairs of images of the same class for comparison, although they didn't address any deformation issues. They used average number of corresponding regions for comparing the detectors and number of matches for descriptors.

A comparison between two patch descriptors (SIFT, SURF) and two binary ones (ORB, FREAK) has been introduced in [20]. This comparison was for Arabic character recognition on mobile devices. Its conclusion was that binary descriptors, especially ORB, yield similar results in terms of characters matching performance to the famous SIFT. However, they are faster in computation and more suitable for mobile applications. Figat et al. [21] have evaluated the performance of image features with binary descriptors. They have investigated the performance with different deformations and with different detector/descriptor combinations. Their results show that the combination of ORB detector with FREAK descriptor outperforms other combinations.

III. EVALUATION FRAMEWORK

A. Performance criteria

The evaluation procedure of [14] is applied in this

work. It's based on the number of correct and false matches between features extracted within an image pair. We use the number of features extracted, repeatability score as well as time for the detection stage and precision and recall for description stage.

Repeatability of a detector is the percentage of features detected again in a transformed query image.

$$Repeatability = \# \, correspondences \, / \, min \, (n1, n2) \quad (1)$$

Where *correspondences* are the overlapped features between two images by a percentage of 50% using a predefined homography H and *n*1 and *n*2 are the number of features detected in the query and train image respectively.

Precision is the number of correct matches relative to the number of all matches between features of two images.

$$Precision = \# \, correct \, matches \, / \, \# \, matches \quad (2)$$

To obtain the number of matches, each feature descriptor in the query image is matched with all other features descriptors in the train image in a Brute-force manner. There is a match between two features with minimum Hamming distance. Symmetry test is then applied to the found matches to obtain only good matches. This keeps only matches between pairs (i, j) such that for i-th query descriptor the j-th descriptor in the matcher's collection is the nearest and vice-versa otherwise, the match would be rejected. A match between two features (a, b) is correct if the error of the area covered by them is less than 30% of the region union and the difference between the first feature and the transformation of the second one by a homograhpy H is less than 3 pixels [14].

$$|| \, a - Hb \, || < 3 \quad (3)$$

For more consistency, the RANSAC (Random Sample Consensus) [22] algorithm is used to keep the inliers and reject all outliers. It iteratively fits a mathematical model to a selected random subset of hypothetical inliers. Finally it produces either a model which is rejected because too few points are part of the consensus data set, or a refined model together with a corresponding consensus set size. It has been widely used for this purpose. For example, Almeida et al. [23] used SIFT algorithm and RANSAC for creating a super-resolution image from a sequence of images with application of character recognition.

Recall is the number of correct matches relative to the number of correspondences.

$$Recall = \# \, correct \, matches \, / \, \# \, correspondences \quad (4)$$

B. Data Set

Our evaluation is carried out on images of seven object classes chosen from the ImageNet database [24]. It is a large-scale data set of images built upon the backbone of the WordNet structure. All the test images are 375×500 pixels or vice-versa and are firstly converted into gray scale. The image sequences which represent the object classes used are *leopard*, *books*, *buildings*, *cups*, *faces*, *trees* and *watches*. Samples of them are shown in Fig. 1. The average entropy of images of each sequence is shown in Table 1. Each image sequence consists of 60 images, ten reference images and five artificial deformations of each image. The deformations are 2D rotation of 45 °, uniform scale with a factor of 1.5, uniform reduction in the brightness with a value of 100, adding a Gaussian noise with standard deviation (σ) equal to 10 and adding a Gaussian blur with standard deviation equal to 1. In our experiments, for each sequence, features are extracted for all images and described. Then, a matching process is carried out between feature vectors in the reference image and every deformed one. The overall results for every sequence are reported.

Fig.1. Test images samples.

Table 1. Average entropy values.

Object Class	Entropy
Leopard	7.6
Books	7.0
Buildings	7.2
Cups	6.9
Faces	7.3
Trees	7.6
Watches	6.1

C. Implementation Details

OpenCV [25] implementations for all detectors and descriptors are used. All the experiments have been carried out on a 2.13GHz/3MB cache Intel® Core™ i3 330M, 2GB RAM, x64 Windows 7, the code is compiled by the Microsoft Visual C++ 2010 Express with OpenCV-2.4.5.

IV. EXPERIMENTAL RESUTLS

In this section, the results of the performance evaluation of recent detectors and descriptors for images of different object classes are discussed. First, features are extracted in all the images by the detectors and the average number of features and the average extraction.

Repeatability is calculated as (1) for every image pair (the reference image and the deformed one) and the

average repeatability is reported for every sequence. Next, those features are described with different descriptors. A Brute-Force matcher is used to match the feature vectors for every image pair and all the matches are checked to calculate the number of correct matches.

Equations (2) and (4) are used to calculate precision and recall respectively for every image pair. Then the average precision and recall are reported for every sequence. All the features are obtained with default parameters of their implementation.

A. Comparison of detectors

The average number of extracted features for every sequence is shown in Fig. 2. BRISK is greater than ORB only for leopard and trees. For all sequences, FAST extracts the largest number of features. In [16], the number of features extracted by SIFT was less than those extracted by SURF. However, this is true here for all sequences except for leopard and trees which have the highest variability in their visual appearance and the highest entropy values as shown in table 1.
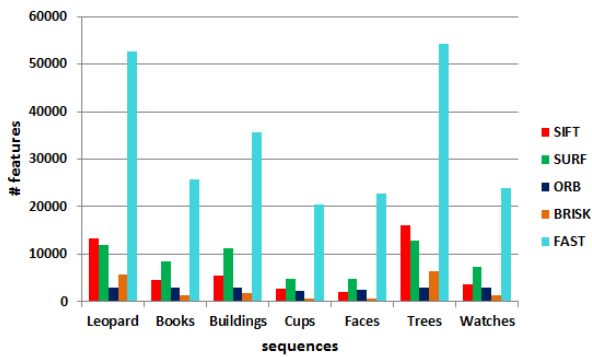

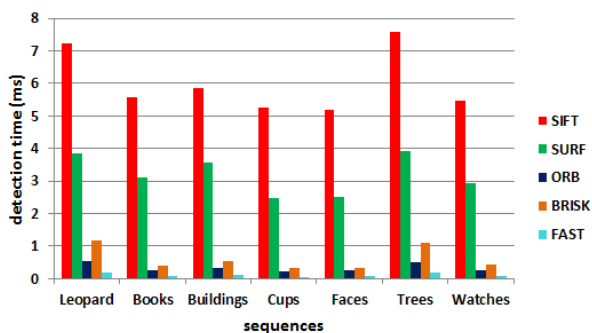Fig.2. Average number of features.

Fig.3. Average detection time in milliseconds.

For detection time, the results are the same of [16] for all sequences, although the order of extracted features numbers is different. As shown in Fig.3, SIFT is the slowest and FAST is the fastest one. There's a big difference in detection time between SIFT and SURF and the rest.

Fig.4 presents the repeatability scores which measure the accuracy of detectors. SUFR and ORB have the best results with small difference for all sequences except for *leopard* and *trees*. Note that BRISK works better for

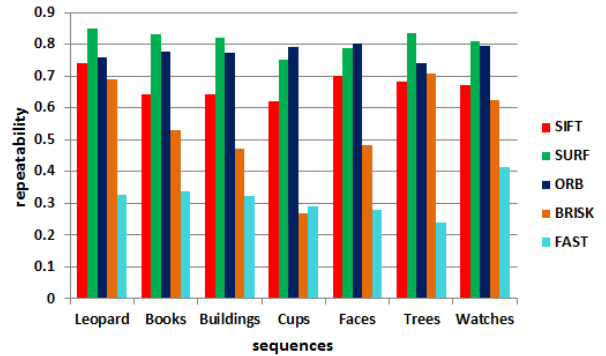*leopard* and *trees*. Unlike [16], FAST is the worst one.
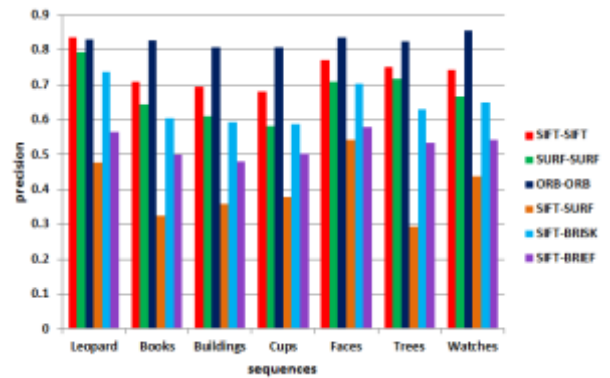

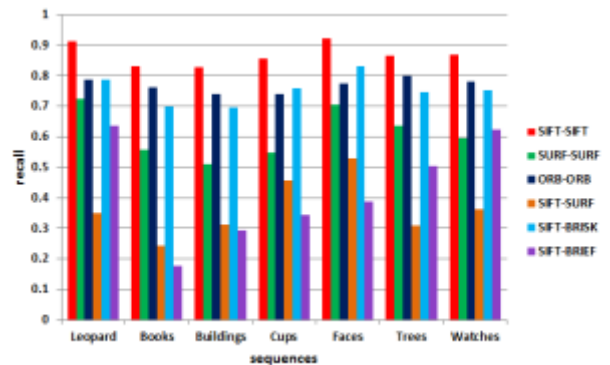Fig.4. Average repeatability.

Fig.5. Average precision

Fig.6. Average recall

B. Comparison of descriptors

In order to compare between descriptors, combinations are made between different detectors and descriptors. Binary descriptors are matched using a Brute-force matcher with efficient evaluation of the Hamming distance. Precision and recall are used for evaluation as shown in Fig. 5 and Fig. 6 respectively.

ORB and SIFT outperform the other descriptors for most sequences in terms of precision and recall. ORB outperforms SURF, although the difference between them decreases for *leopard*, *trees* and *watches*. This means that SURF works better for classes with high variability in their visual appearance.

Moreover, SURF descriptor works better with SURF features other than with SIFT features. Using SIFT detector, BRIEF is better than SURF in terms of recall for some sequences.

V. CONCLUSION

In this paper, we have presented an evaluation of local feature detectors and descriptors with different deformations on a set of images of different classes and structures. The goal was to demonstrate if and how different detectors and descriptors deal with images of different structures with the same performance level. We have used an evaluation protocol which was previously used in other evaluations. For detectors comparison FAST is the fastest detector, although it has the greatest number of extracted features for all image classes. The order of features extracted by the SIFT, SURF, ORB and BRISK differs depending on the image structure. Similarly, the order of detectors repeatability isn't the same for different classes. For descriptors comparison, the results were more stable. ORB and SIFT obtain better results than the other descriptors for most classes, however the differences between them varies depending on the image structure. It would be interesting to evaluate feature detectors and descriptors on a wide variety of image structures in order to generalize a rule for choosing the best method suitable for a certain image structure.

REFERENCES

[1] D. G. Lowe, "Object recognition from local scale-invariant features," *The Proceedings of the Seventh IEEE International Conference on Computer Vision,* vol. 2, pp. 1150-1157, 1999.

[2] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.,* vol. 38, no. 4, pp. 1-45, 2006.

[3] M. Brown, and D. G. Lowe, "Unsupervised 3D object recognition and reconstruction in unordered datasets," *Proceedings of the international conference on 3D digital imaging and modeling,* pp. 56-63, 2005.

[4] N. Snavely, S. M. Seitz, and R. Szeliski, "Skeletal graphs for efficient structure from motion," *Proc. Computer Vision and Pattern Recognition,* 2008.

[5] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *International Journal of Robotics Research,* vol. 21, no. 8, pp. 735- 758, August 2002.

[6] T. Tuytelaars, and K. Mikolajczyk, "Local invariant feature detectors: a survey," *Found. Trends. Comput. Graph. Vis.,* vol. 3, no. 3, pp. 177-280, 2008.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.

[8] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding,* vol. 110, no. 3, pp. 346-359, 2008.

[9] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," *Proceedings of the IEEE International Conference on Computer Vision,* pp. 2564-2571, 2011.

[10] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: binary robust independent elementary features," *Proceedings of the 11th European Conference on Computer Vision,* pp. 778-792, 2010.

[11] E. Rosten, and T. Drummond, "Machine learning for high-speed corner detection," *Proceedings of the 9th European Conference on Computer Vision,* pp. 430-443, 2006.

[12] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: binary robust invariant scalable keypoints," *Proceedings of the International Conference on Computer Vision,* pp. 2548-2555, 2011.

[13] K. Mikolajczyk, et al., "A Comparison of Affine Region Detectors," *International Journal of Computer Vision,* vol. 65, no. 1-2, pp. 43-72, 2005.

[14] K. Mikolajczyk, and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 27, no. 10, pp. 1615-1630, 2005.

[15] L. Juan, and O. Gwun, "A comparison of SIFT, PCA-SIFT and SURF," *International Journal of Image Processing,* vol. 3, no. 4, pp. 143-152, 2009.

[16] O. Miksik, and K. Mikolajczyk, "Evaluation of local detectors and descriptors for fast feature matching," *Proceedings of the International Conference on Pattern Recognition,* pp. 2681-2684, 2012.

[17] D. Gossow, P. Decker, and D. Paulus, "An evaluation of open source SURF implementations," *J. Ruiz-del-Solar, E. Chown, and P. Plöger, ed., RoboCup 2010: Robot Soccer World Cup XIV, Springer,* pp. 169-179, 2011.

[18] J. Bauer, N. Sünderhauf, and P. Protzel, "Comparing several implementations of two recently published feature detectors," *Proceedings of the International Conference on Intelligent and Autonomous Systems,* 2007.

[19] J. Lankinen, V. Kangas, and J.-K. Kamarainen, "A comparison of local feature detectors and descriptors for visual object categorization by intra-class repeatability and matching," *International Conference in Pattern Recognition,* 2012.

[20] M. Tounsi, I. Moalla, A. M. Alimi, and F. Lebourgeois, "A comparative study of local descriptors for Arabic character recognition on mobile devices," *Seventh International Conference on Machine Vision,* vol. 9445, 2015.

[21] J. Figat, T. Kornuta, and W. Kasprzak, "Performance evaluation of binary descriptors of local features," *ICCVG, LNCS,* vol. 8671, pp. 187-194, 2014.

[22] M.A. Fischler, and R.C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM,* vol. 24, no 6, pp. 381-395, 1981.

[23] L. L. Almeida, M.S. Paiva, F. A. Silva, and A. O. Artero, "Super-resolution image created from a sequence of images with application of character recognition," *International journal of Intelligent Systems and Applications,* vol. 6, no 1, pp. 11-19, 2013.

[24] J. Deng, et al., "Imagenet: a large-scale hierarchical image database," *IEEE Conference on Computer Vision and Pattern Recognition,* pp. 248-255, 2009.

[25] G. Bradski, "The opencv library," *Doctor Dobbs Journal,* vol. 25, no. 11, pp. 120-126, 2000.

Authors' Profiles

Faten A. Khalifa was born on March 8, 1987 in El-Behira, Egypt. She received the B.Sc. degree in information technology from Faculty of Computers and Information, Menoufia University, Menoufia, Egypt in 2008 with grade very good with first honor degree.

During 2008-2010, she was studying pre-masters in the information systems department in Faculty of Computers and Information, Menoufia University, Menoufia, Egypt. She has registered master in real time markerless tracking for augmented reality applications. Since 2009 till now, she is a demonstrator at the Faculty of Computers and Information, Menoufia University, Menoufia, Egypt. Her research interests are computer vision, image processing, object recognition, multimedia, mobile applications, virtual and augmented reality. Her email is: faten_a_khalifa@ci.menofia.edu.eg.

Noura A. Semary works as an associated prof. in Faculty of Computers and Information, Menofia University, Egypt. She has BSc in 2001 from Cairo University, Faculty of Computers and Information. She worked as a staff member in the Faculty of Computers and Information, Menofia University, Egypt in 2003. In 2007 and 2011 she has obtained her Master and Ph.D. degrees respectively in Information Technology from Computers and Information Faculty at Menofia University. Her main research interests are in Image Processing, Computer Vision, Data Compression, Data Hiding, Virtual Reality and Assistive Technology fields. In 2009 she won the first rank in "Made in Egypt" and "Made in Arab World" competitions for her project "Black and White Movies Colorizer". In 2012 she has joined The Scientific Research Group in Egypt (SRGE). She has 25 publications in international conferences and journals. Also, she has reviewed more than 10 papers for international conferences and journals in 2014. She is also leading and supervising various graduation projects and post-graduate students.

Hatem M. El-Sayed obtained his B.S. and M.SC. (by research) both in Electrical Engineering from the Alexandria University , Faculty of Engineering , Egypt in 1990 and 1995 respectively. He obtained his Ph.D. degree in Electrical Engineering also from Alexandria University, Faculty of Engineering, Egypt in 2001 specializing in neural networks and applications. He is currently a professor in Information systems department, Faculty of Computers and Information, Information systems department, Faculty of Computers and Information, Menoufiya University, Egypt since 2004. He has worked on a number of research topics and consulted for a number of organizations. He has contributed more than 30+ technical papers in the areas of neural networks, Database applications, Information security and Internet applications.

Mohiy M. Hadhoud is a professor of digital communications, signals, image, and multimedia processing. He obtained his Ph.D from the Department of Electronics and Computer Science, University of Southampton, Southampton, England, in 1987. He worked as Vice president of Menoufia University for education and student affairs, Dean of the faculty of Computers and Information, Vice dean, for education and student affairs, faculty of Computers and Information, and Head of Information Technology Department, the faculty of Computers and Information. Now, he is on leave as: Dean of the school of Engineering and Business technology, Canadian International College (CIC) - New Cairo. From 1990 to 1996 he worked as academic supervisor and training Expert in the switching department, Telecommunications College, Saudi Telecom, Kingdom of Saudi Arabia. From 1996 until 2001 he worked in the faculty of Electronics Engineering, Department of Electronics and Communications as an assistant professor of Digital communications and signal processing. In 2001 he become a Professor of Digital communications and signal processing. Since 2001 worked as a Professor of information technology and Digital communications and signal processing in the department of Information technology, faculty of computers and Information, Menoufia university. During his long career, Prof. Hadhoud had been responsible for teaching of a large number of courses in the fields of communications and telecommunications engineering, signal and image processing, and information technology. He is a coauthor of the book titled "Image Super-Resolution and Applications " Fathi E. AbdEl-Samie, Mohiy M. Hadhoud, Said E. El-Khamy, CRC press, 2013, by Taylor & Francis. Prof. Hadhoud, research interests are in the areas of digital signal processing, adaptive filtering and image processing, multimedia applications, Information security and data hiding, face recognition, and image and video colorization. He, in cooperation with colleagues and students, has published more than 182 papers in international journals, conferences, local journals and local conferences.