Image Forgery Detection using FREAK Binary Descriptor and Level Set Segmentation

Bayumy A.B. Youssef[1] and Essam H. Atta[2]
Informatics Research Institute (IRI), City of Scientific Research and Technological Applications
(SRTA-City), Alexandria, Egypt
Bbayumy@yahoo.com[1], Essam_atta@yahoo.com [2]

Abstract: This paper describes a novel approach for copy-move image forgery detection using FREAK binary descriptor and level set segmentation. Binary descriptors are fast to compute and compact to store, since they depend only on image intensity comparison. In the proposed approach image, key features are computed using FAST detector and feature matching is implemented using hamming distance. Clustering of matched features is achieved by using region-based Level set segmentation. Experimental results carried on several image data sets and comparison with SIFT based methods show that the proposed approach is accurate and effective in terms of copy move forgery identification and computational speed.

Keyword: Copy-move Image forgery, Binary descriptors, Level set, Image Segmentation

1. Introduction

Detection of digital Image forgery is a relatively new scientific field that addresses the problem of digital images authenticity. Advances in digital technologies have made it easy to manipulate and counterfeit digital images. This problem is aggravated by the availability of vast amount of images available through the internet [1-5]. Detecting image forgery has significant implications in a variety of applications such as copyright protection, proof of ownership, legal, commercial, and security related issues. One of the most widely known image forgery methods is the copy-move approach, where part of an image is copied and then moved to a different location in the image. This may also be accompanied by a change of scale or rotation. Several methods have been used in the literature to address the copy move detection problem [6]. Most of these methods can be classified as block-based or key point based. In block based methods the image is divided into a number of blocks and matching between blocks determines the copied regions. Key point based methods on the other hand, are based on matching detected key point features such as SIFT/Surf to find forged regions in an image. For example, in references[7-8] SIFT features and J-Linkage clustering algorithms are used for copy-move forgery detection and localization. Other methods including Block type approach, DCT, PCA and others have also been used. A more complete review of these methods can be found in reference[6]. In the paper, a novel approach based on Freak binary descriptor combined with level set segmentation is used for detecting copy-move image forgery. The following sections describe the implementation of the proposed approach.

2. Proposed method

The proposed method can be summarized in four steps as follows:

Step-1: Key points detection

Key feature points are detected using FAST (Features from accelerated segment test) corner detection method.

Step-2: Feature Description and matching

Fast Retina Keypoint (FREAK) binary descriptor and hamming distance metric are used for feature description and matching respectively.

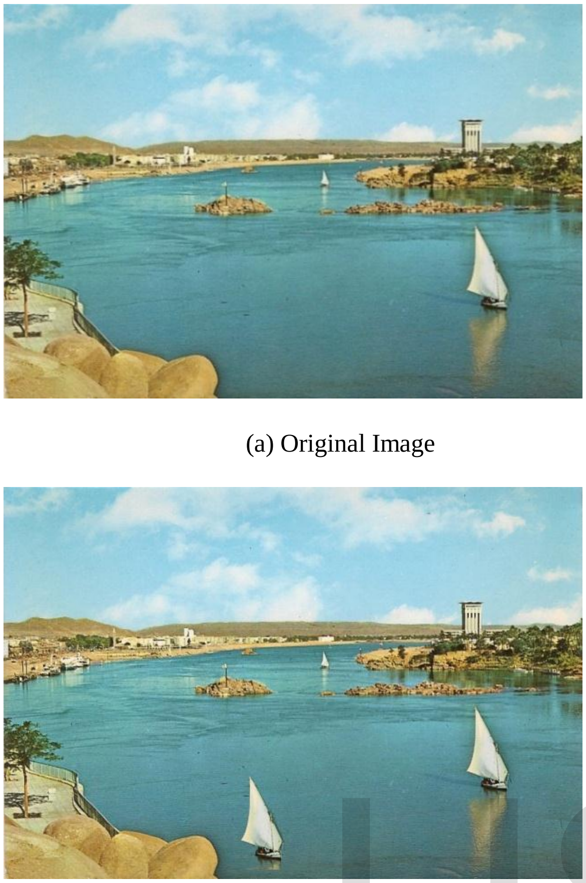
Step-3: Level set Segmentation

Level set method is implemented in the in a region of the image defined by the maximum and minimum coordinates of the matched key points

Step-4: Tampered regions identification

Regions obtained by the level set method are searched using binary mask to determine if they contain three or more matched key points, thus identifying a forged regions.

Details of these steps are explained for the sample test image shown in Fig.1. The original image depicts a sail boat in a natural scene, the sail boat is copied and scaled then moved in a different location in the image. The tampering detection steps are as follows:



(a) Original Image

(b) Fake Image

Fig. 1: Original and fake images

2.1. Step 1: Key point detection using (FAST)

Key point detection is obtained using Features from accelerated segment test (FAST) method. This is a computationally efficient corner detection method [9, 10], which could be used to extract feature points in images. FAST is faster than many other well-known corner detection, such as Harris, SUSAN and DOG methods [11]. In Fast, a point “P” is classified as a corner based on finding a sufficiently large set of n contiguous pixels in a circle of 16 pixels around that point (Fig.2); such that these pixels are all brighter than (I_p +t) +t or darker than (I_p-t) –t, where I_p is the intensity at p and t is a threshold value. In the present approach n and t are set to 9, and 30 respectively. Non-maximal suppression is also used to refine the detection process, when detecting multiple interest points in adjacent locations. A score function, v is computed for all the detected feature points. is the sum of absolute difference between p and 16 surrounding pixels values is computed for all the detected feature points and is used as a score function. For two adjacent keypoints the one with lower score function is discarded. The keypoints detected using FAST algorithm for the sample test case of Fig.1 is displayed in Fig.3.

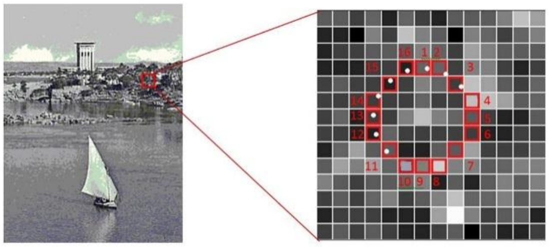


Fig.2: FAST feature detection layout [9]

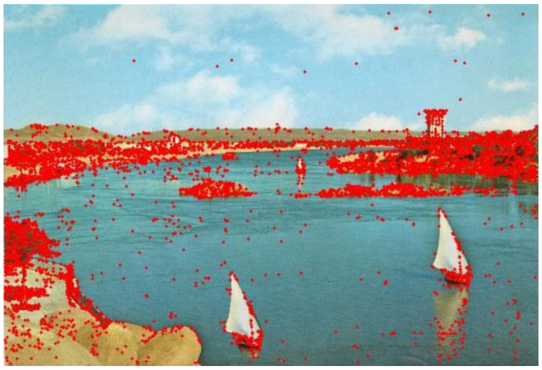


Fig. 3: Keypoints detected using FAST method

2.2. Step 2: FREAK Binary Feature descriptor

Once the keypoints are obtained a feature descriptor is constructed using Fast Retina binary descriptor (FREAK).Binary feature descriptors such as BRISK, ORB, and FREAK [12-18] have emerged as a viable option to gradient based descriptors such as SIFT and SURF [19, 20, 21]. Binary features are fast to compute and compact to store. They encode the features in an image patch as a binary string using only comparison of image intensity. Thus, fast matching between two patch descriptions can be achieved using the hamming distance as a similarity measure between two binary strings. Fast Retina Keypoint (FREAK) is a binary descriptor based on human retina visual system [18] that presents the best trade-off between performance and speed [7]. In general, all binary descriptors require a sampling pattern; FREAK sampling pattern is displayed in Fig. 9 exhibiting rings of points around a center with different levels of Gaussian Smoothing. However, much larger difference in smoothing levels for the outer rings than the inner rings and the points share overlapping data. Its spatial resolution is fine near the center and becomes coarser when moving away from it. Sample points located around the given keypoint are smoothed with a Gaussian kernel that is varies with respect to the location of the sampling point to simulate the behavior of the human retina. The smoothed areas around the sampling points are referred to as receptive fields [18]; and the centers of the receptive fields represent the sampling points of the FREAK descriptor. The descriptor is obtained by concatenating simple pairwise intensity comparison tests, pairs are selected by evaluating 43 weighted Gaussians at locations concentrated around the keypoint [18] as shown in Fig. 9. The descriptor *F* can then be expressed as:

$$F = \sum_{0 \le a \le N} 2^a T(P_a) \qquad (1)$$

Where

$$T(P_a) = \begin{cases} 1 & if\,(I(p_a^{r_1}) - I(p_a^{r_2})) > 0 \\ 0 & otherwise \end{cases} \qquad (2)$$

Moreover, $I(p_a^{r_1})$ is the smoothed intensity of the first receptive field of the pair p_a.

A cascade is used for comparing the selected pairs, where the 64 most important bits are placed in the beginning of the descriptor, thus, improving the speed of the comparison process.

To account for rotation invariance, FREAK generates a 45 points sampling pattern specifically for orientation, as shown in Fig.5, where the blue circles are sampling points, red circles are smoothed receptive fields, and lines represent keypoint pairs. This pattern is symmetric and is used to generate the local gradient, thus the orientation *O* of a given keypoint can be computed as:

$$O = \frac{1}{M} \sum_{p_o \in G} (I(p_o^{r_1}) - I(p_o^{r_2})) \frac{I(p_o^{r_1}) - I(p_o^{r_2})}{\left\| I(p_o^{r_1}) - I(p_o^{r_2}) \right\|} \qquad (3)$$

Where G is the set of all the pairs used to compute the local gradients is the number of pairs in G and $p_o^{r_i}$ is the 2D vector of the spatial coordinates of the center of receptive field.

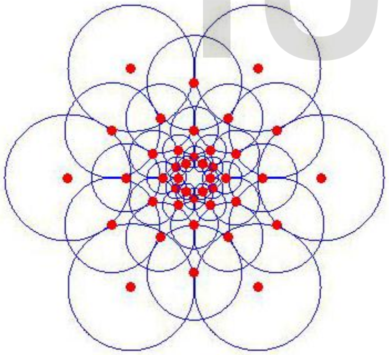


Fig. 4: FREAK sampling pattern [18].

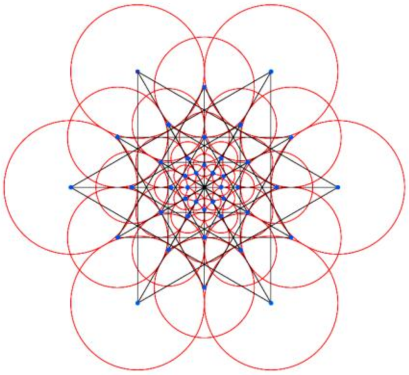


Fig.5: FREAK orientation pattern [18]

Matching of FREAK features is performed using the hamming distance as a similarity metric, and can be computed efficiently by bitwise XOR followed by bit count. Implementation of FREAK feature extraction and matching will produce a list of matched pairs of points. Fig.6 display the matched pairs of keypoints for the test case of Fig.3. A clustering step is then performed to find which region in the image contains the matched pairs, clusters that contain three or more matched pairs indicated a forged region in the image. The clustering step is implemented using the level set method.

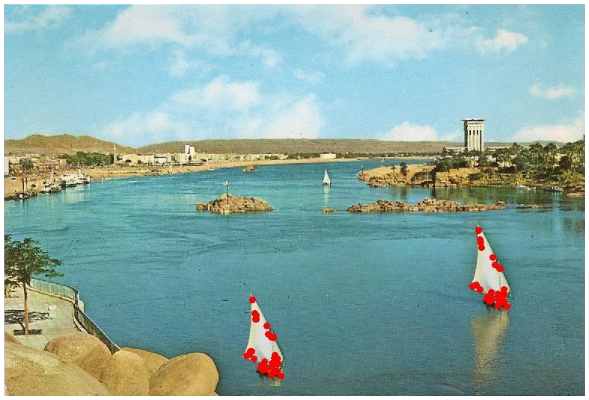


Fig. 6. FREAK matched key points

2.3. Step 3: Level Set Segmentation

The level set method originally introduced in references [22 -24] has been applied to a wide variety of many applications. In this paper, the variational level set formulation [25] is used for image segmentation. This variational formulation is computationally more efficient than the traditional level set methods and has the advantage of eliminating the costly reinitialization step in the level set procedure. In this formulation the level set function is obtained by solving the following evolution equation:

$$\begin{aligned} \frac{\partial \phi}{\partial t} = \mu \left[\Delta\phi - div\left(\frac{\nabla\phi}{|\nabla\phi|} \right) \right] \\ + \lambda\,\delta(\phi)\,div\left(g\frac{\nabla\phi}{|\nabla\phi|} \right) \qquad (4) \\ + \nu g\,\delta(\phi) \end{aligned}$$

Where $\phi, \delta, \Delta, \nabla$ are the level set function, the Dirac function, the Laplacian, and gradient operator respectively. μ, λ, and ν are constants and, g is an edge indicator function defined as:

$$g = \frac{1}{1 + [\nabla G_\sigma * I]^2} \qquad (5)$$

Where G_σ is the Gaussian kernel with standard deviation σ and I is the image intensity.

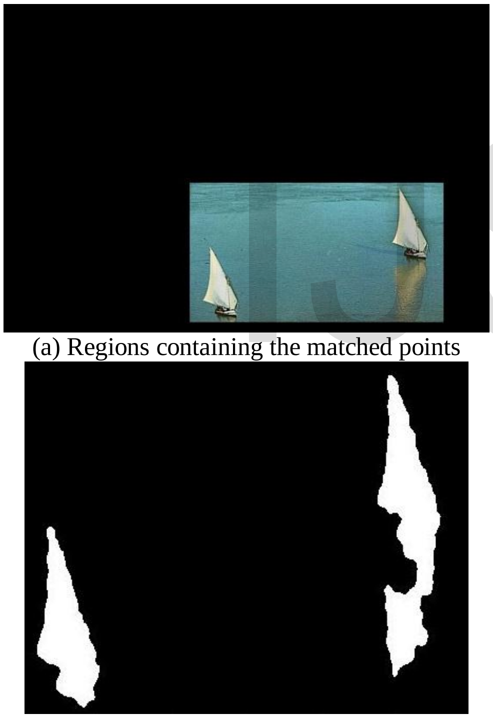
Equation (5) represents a gradient flow evolution where the first term controls gradient diffusion while the second and third terms drive the zero level set function towards the object boundaries.

Numerical solution of equation (5) is obtained explicitly using central and forward differences for the spatial and time derivatives respectively as follows.

$$\phi_{i,j}^{k+1} = \phi_{i,j}^{k} + \tau L(\phi_{i,j}^{k}) \qquad (6)$$

Where τ is the time step, and $L(\phi_{i,j}^{k})$ is the finite difference approximation of the right hand side of equation (5). Values used for the time step and constants in equation (5) are the values recommended in [19], namely. $\nu = 1.5$, τ =5, and $\mu * \tau \le 0.25$.

By using the coordinates of the matched keypoints obtained in the previous step, a region is cut from the image Fig.7 (a) and the level set method is used to segment this region into a number of clusters, thus identifying the different objects in the cut region as displayed in Fig.7 (b).

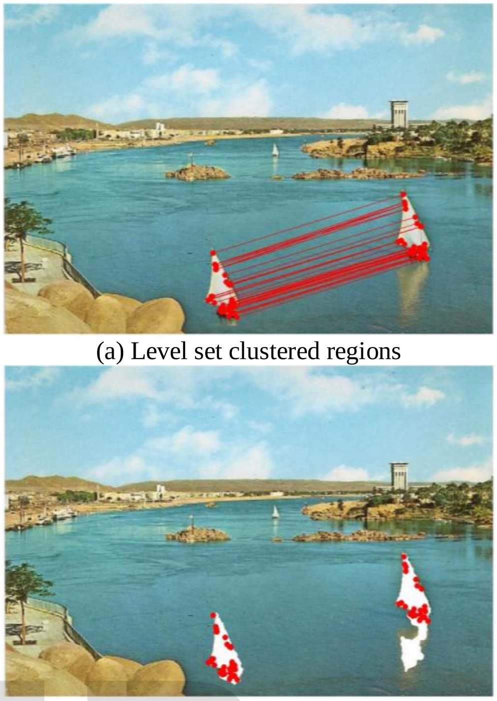

(a) Regions containing the matched points

(b) Level set segmentation

Fig.7: Level set segmentation

2.4. Step 4: Keypoints Regions Identification

A binary mask is used for the segmented regions found, and the number and labeling of the segmented regions are found using morphological operations. A square 3x3 pixels structuring element combined with dilation is used to label and extract the segmented components found by the level set method, this is followed by a check and search procedure to find which region contains each matched keypoint. If three or more matched key points are found in any single region then the image is identified as a forgery. Fig.8 (a) and (b) display the two-clustered objects containing the matched keypoints, which indicate that the image has been tampered with.


(a) Level set clustered regions

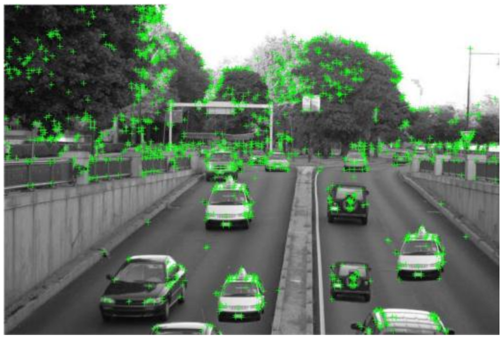
(b) Tampering detection

Fig.8: Identification of copied objects

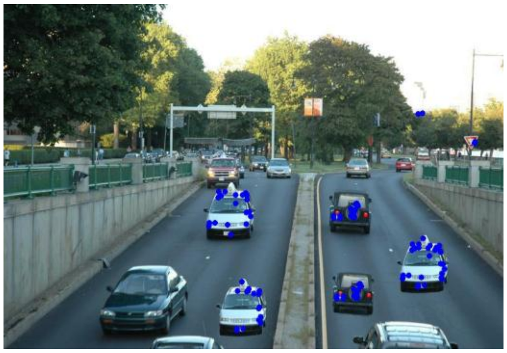
3. Experiments

The developed approach is tested for number of cases to assess its capability for image forgery detection. Fig. 9 displays the results for a 800x552 pixels tampered image taken from reference [7] which have multiple copies of the same region. The level set segmentation produces eight clusters, and the outcome of the search process resulted in three empty clusters and five clusters that contain the matched keypoints, thus forgery is identified. The results are compared with the Sift based approach of reference [7], the number of matched keypints found is less than those obtained with the sift based method, however, the present approach successfully identified the copied regions. The level set segmentation requires 1200 iterations to converge and proved to be effective in capturing the different objects present in the regions containing the matched keypoints. The computations are implemented on 1.6GHz processor, and it took 14.92 seconds. The computational time required for the level set segmentation is greatly reduced since only part of the image is segmented

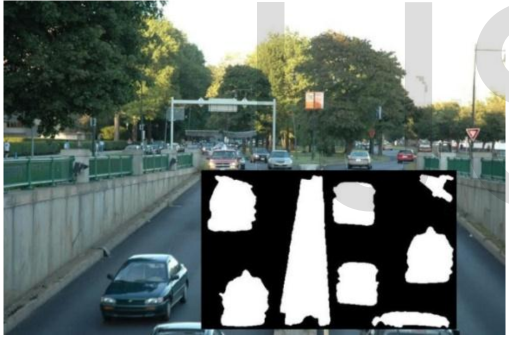
(a) Forged image



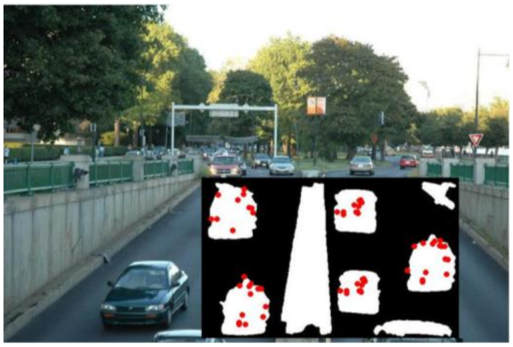
(b) FAST detected key points



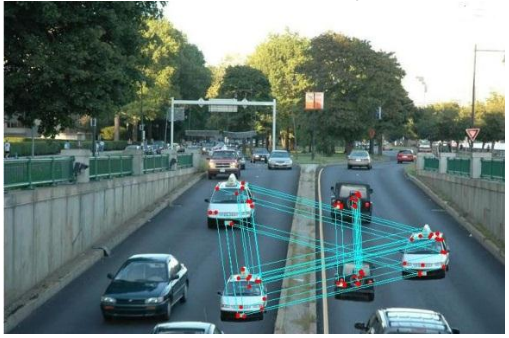
(c) Matched key points



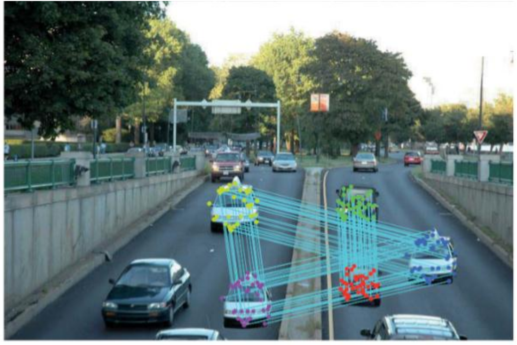
(d) Segmented regions



(e) Level set clustered regions



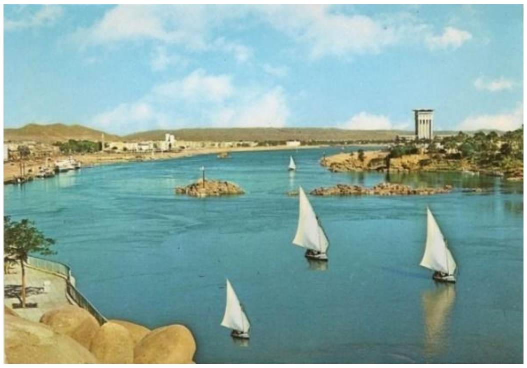
(f) Tampering detection



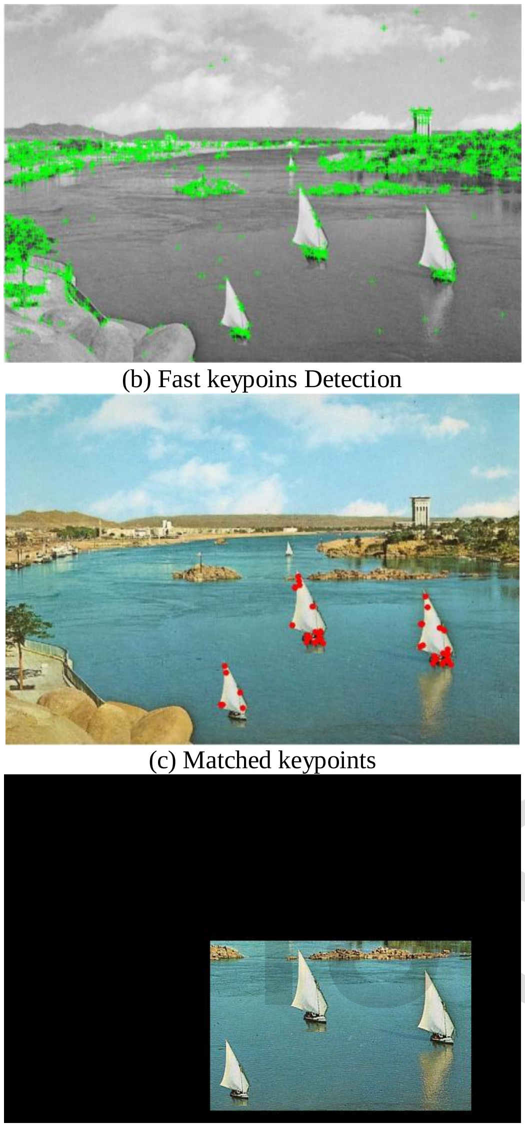
(g) Result of Sift based approach [7]

Fig.9: Multiple object copy move case

Another case involving multiple copied regions in addition to scale change for 1165x788 pixels image shown is displayed in Fig.10. Two copies of the sail boat, one of which is scaled down are placed in different locations in the image. It should be noted that the level set computation time is dependent on the size of the region containing the matched key points.



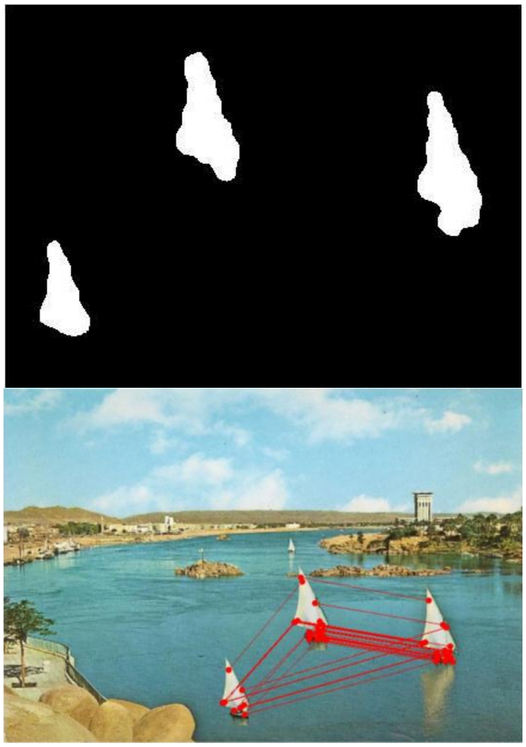
(a) Foreged Image



(b) Fast keypoins Detection

(c) Matched keypoints

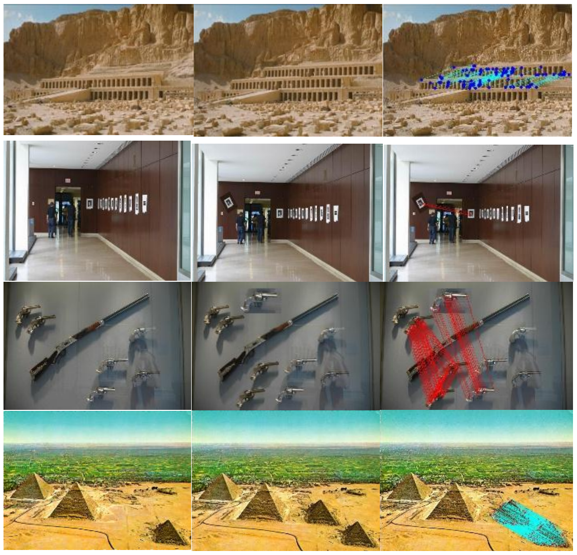
(d) Region cut for segmentation



(e) Level set segmented objects (f) Tampering detection

Fig.10: Case of multiple object copy move with scaling

Additional test cases examples of carefully forged images are shown in Fig.11, where the original image, the forged one, and the detection results are shown in the first, second, and third columns respectively. Copy move forgery with rotation is displayed in the second row. Images in the second and fourth rows are taken from reference [7] .The results for all the cases considered demonstrate the effectiveness of the developed approach in identifying copy move image tampering.



a: original b: copy move forgery c: Tampering detection

Fig.11: Addition test cases

4. Conclusion

In this paper, a new approach based on FREAK binary features descriptor and level set segmentation is developed to detect copy-move image forgeries. The introduction and use of binary features combined with level set segmentation proved to be an effective way for identifying forged regions in an image and presents a competitive solution to other SIFTS or SURF based methods. Experimental tests have been carried out on different datasets containing various typologies of fake images and also original ones. Results confirm that the proposed method outperforms other similar state-of the- art techniques both in terms of copy-move forgery detection reliability and of precision in the localization of the manipulated objects.

References

[1]. Fridrich, J. ; “Digital Image Forensics” , Signal Processing Magazine, IEEE, Vol. 26 , Issue 2 ,Pp 26–37, 2009.

[2]. Tran Van Lanh; Kai-Sen Chong; Emmanuel, S.; Kankanhalli, M.S., “A Survey on Digital Camera Image Forensic Methods”, IEEE International Conference on Multimedia and Expo,

[3].

[4]. Hany Farid, “Digital Image Forensics”, SCIENTIFIC AMERICAN, INC., 2008.

[5]. H. T. Sencar and N. Memon, “Overview of State-of-the-Art in Digital Image Forensics”, Pp.13-36, WSPC – Proceedings, 2007.

[6]. Alessandro Piva, “An Overview on Image Forensics”, SRN Signal Processing
Volume 2013, Article ID 496701, 22 pages, 2013.

[7]. Salam A. Thajeel and Ghazali Bin Sulong, “State of the Art of Copy-Move Forgery Detection Techniques: A Review”, IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 6, No. 2, November 2013.

[8]. I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, G. Serra. “A SIFT-based forensic method for copy-move attack detection and transformation recovery”, IEEE Transactions on Information Forensics and Security, Vol. 6, issue 3, pp. 1099-1110, 2011.

[9]. Irene Amerini, Lamberto Ballan, Roberto Caldelli, Alberto Del Bimbo, Luca Del Tongo, Giuseppe Serra “Copy-move forgery detection and localization by means of robust clustering with J-Linkage”, Signal Processing: Image Communication, Vol. 28, Issue 6, Pp. 659–669, July 2013.

[10]. Edward Rosten and Tom Drummond, “Machine learning for high speed corner detection” in 9th European Conference on Computer Vision, vol. 1, Pp. 430–443, 2006

[11]. Edward Rosten, Reid Porter, and Tom Drummond, “Faster and better: a machine learning approach to corner detection” in IEEE Trans. Pattern Analysis and Machine Intelligence, Vol 32, Pp. 105-119, 2010.

[12]. T. Tuytelaars and K. Mikolajczyk, “Local Invariant Feature Detectors: A Survey”, Foundations and Trends in Computer Graphics and Vision, Vol. 3, No. 3, Pp. 177–280, 2007

[13]. Schaeffer, Cameron, “A Comparison of Keypoint Descriptors in the Context of Pedestrian Detection: FREAK vs. SURF vs. BRISK”, 2013.

[14]. Janez Kriˇzaj, et al. “SIFT vs. FREAK: Assessing the usefulness of two keypoint descriptors for 3D face verification”, Information and Communication Technology, Electronics and Microelectronics (MIPRO), 37th International Convention on, 2014.

[15]. Janez Krizaj, Vitomir Struc, France Mihelic, “A Feasibility Study on the Use of Binary Keypoint Descriptors for 3D Face Recognition”, 6th Mexican Conference, MCPR Proceedings, Springer International Publishing, pp. 142-151, Cancun, Mexico, June 25-28, 2014.

[16]. Rublee, Ethan, et al. “ORB: an efficient alternative to SIFT or SURF”, IEEE International Conference on Computer Vision (ICCV). IEEE, 2011.

[17]. Calonder, Michael, et al. “Brief: Binary robust independent elementary features.” Computer Vision–ECCV. Springer Berlin Heidelberg. Pp. 778-792, 2010.

[18]. Leutenegger, Stefan, Margarita Chli, and Roland Y. Siegwart. “BRISK: Binary robust invariant scalable keypoints”, 2011 IEEE International Conference on Computer Vision (ICCV). IEEE, 2011.

[19]. Alahi, Alexandre, Raphael Ortiz, and Pierre Vandergheynst. “Freak: Fast Retina Keypoint.” IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2012.

[20]. Lowe, David G. “Object Recognition from Local Scale-Invariant Features.”, proceedings of the seventh IEEE international conference on Computer vision. Vol. 2. IEEE, 1999.

[21]. Bay, Herbert, Tinne Tuytelaars, Luc Van Gool, “Surf: Speeded up robust features”, Computer Vision–ECCV. Springer Berlin Heidelberg. Pp. 404-417, 2006.

[22]. Mikolajczyk, Krystian, and Cordelia Schmid. “A Performance Evaluation of Local Descriptors.” IEEE Transactions On Pattern Analysis And Machine Intelligence‹ Vol. 27, No. 10, Pp. 1615-1630, OCT 2005.

[23]. Osher, S.; Sethian, J. A., “Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations”, Journal of Computational Physics, Vol.79, Pp.12-49, 1988.

[24]. Osher, Stanley J.; Fedkiw, Ronald P., “Level Set Methods and Dynamic Implicit Surfaces”, New York, Inc. Springer-Verlag, Series Volume: 153; 2002.

[25]. Sethian, James A, “Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics”, Computer Vision, and Materials Science. Cambridge University Press. ISBN 0-521-64557-3., 1999.

[26]. Chunming Li, Chenyang Xu, Changfeng Gui, and Martin D. Fox, “Level Set Evolution Without Re-initialization: A New

Variational Formulation" Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) Pp. 1063-6919, 2005.