Detection of Copy-Move Forgery in Images Containing Flipped and Random-Shaped Tampered Regions

Reem A. Alsanussi, Hossam Eldeen Shamardan, Mostafa-Sami M. Mostafa
Computer Science, Information Technology Department, Computer Science Department
Faculty of Computers and Information, Helwan University, Egypt
reem1188@yahoo.com,hossammosh@hotmail.com, mostafa_sami@hotmail.com

Abstract

Copy-move forgery is one of the most common tampering methods. Its main purpose is to hide an object or a region in an original image. This paper proposes a methodology for efficient and robust detection of image forgery for different cases. Our new approach can efficiently detect both squared and random-shaped tampered regions. It can also detect vertically and/or horizontally flipped tampered regions. Moreover, it can detect tampered region containing holes, intersected regions, and multiple copy-move regions. The processes of adding either white Gaussian noise or blurring are very common in digital image tampering. They could be used to conceal the influence caused by copy-move process and to remove unwanted defects. The results showed that our developed detection algorithm can detect tampered regions that are distorted by Gaussian white noise or blurring. The necessary condition for detection of a random-shaped tampered region is determined. Our experiments showed that the accuracy of detection depends on the shape of tampered regions. Different detection algorithms are tested to reach the lowest computational complexity.

Keywords: *Digital image forensics, Region duplication detection, copy-move forgery, Image flipping, kd-tree algorithm, lexicographical sort.*

1. Introduction

Digital images are a widely used medium of communication. They are widely spread on the internet. With the increasing of the power of digital image processing software packages, image forgery is becoming easier. Images have to be trusted and authenticated in order to using them safely in different areas like forensic investigation, intelligence services, surveillance systems, journalism, and medical imaging. The verification of images authenticity became a significant problem. There is a need to find methods to make us trust images obtained from unsecured sources. Image tampering detection is considered as a subfield of the image forgery detection. Detection of image forgery is a branch of image forensics which aims at assessing the authenticity and the origin of images.

Detecting Image forgery is an emerging field of research. Copy-Move is a type of forgery in which a part of an image is copied and then pasted on to another location the same image. In general, the main intentions to use the copy-move forgery are:

- Hiding or removal of objects from the original image.
- Addition of objects in the original image.
- Change of appearance of objects in the original image.

The primary task of a copy-move image forgery detection method is to determine if a given image contains cloned regions without prior knowledge of their shape and location.

In the tampering process, a certain region of the original image is copied and then pasted to another location of the same image. This pasted region is called "*the tampered region*".

There are several approaches to detect tampering in digital images. It can be divided into active and passive-blind approaches [1]. The research area of active approaches can be divided into the digital signature [2] and the data hiding [3]. The data hiding refers to embedding secondary data into the image. One example of this approach is digital watermarking [4]. On the other hand, the blind methods are regarded as a new direction of research works. They do not need any prior information about the image. In this paper, a blind method is developed to detect copy-move forgery in digital images. Many methods have been proposed for the detection of copy-move forgery. Fridrich et. al. [5] first analyzed the exhaustive search and then proposed a block-matching detection method based on the discrete cosine transform (DCT) which proved to be more effective than the exhaustive search. Popescu and Farid [6] applied the method of principal component analysis (PCA) on a small fixed size image blocks to yield a reduced-dimension representation. Shih et. al. [7] compared the performance of DCT and PCA representation in copy-cover forgery detection. Those approaches are based on the concept of the correlation between the original copied region and the pasted region [8]. Huang et. al. [9] proposed an improved method to detect copy-move forgery by reducing the dimension of the extracted feature vectors. However, they failed to consider the situation of multiple copy-move forgery. Cao et al. [10] proposed another scheme, in which each cosine transformed block was represented by a circular block. For each circular block, four feature vectors were extracted. Then, the feature vectors were lexicographically sorted, and duplicated image blocks were determined by a matching process. None of the above methods was directly applied to detect forgery for images involving flipping.

Some attempts to establish a model of region-duplication forgery [11]. Luo's model described four basic constraints of region-duplication forgery, including the region connectivity, two regions un-intersection, translation vector constraint and the duplicate region area threshold. It assumes that the largest copied region must have not any holes inside and be pasted away from its original location without intersecting with its primer location. However, this model could not describe the forgery when one copy region is pasted onto two places. In [12], they performed morphologic operations on the input images to fill the holes in the tampered regions. In this paper, the developed methodology can overcome these problems, since it can detect tampered region containing holes, intersected regions, and multiple copy-move regions.

The goal of this paper is to setup a methodology to detect copy-move forgery in digital images for different cases of the tampered regions including flipped, random-shaped regions. The rest of this paper is organized as follows: in section 2, the proposed methodology is presented. The experimental results are presented in section 3. In section 4, the conclusion is drawn.

2. A Proposed Methodology

A proposed methodology of the detection of the copy-move forgery for different cases is presented in this section. By a methodology, we mean a guideline system for solving the forgery detection problem, with specific components such as stages, tasks, methods, techniques and tools. Different software tools are developed to process images through different experimental stages as shown in Figure (1).

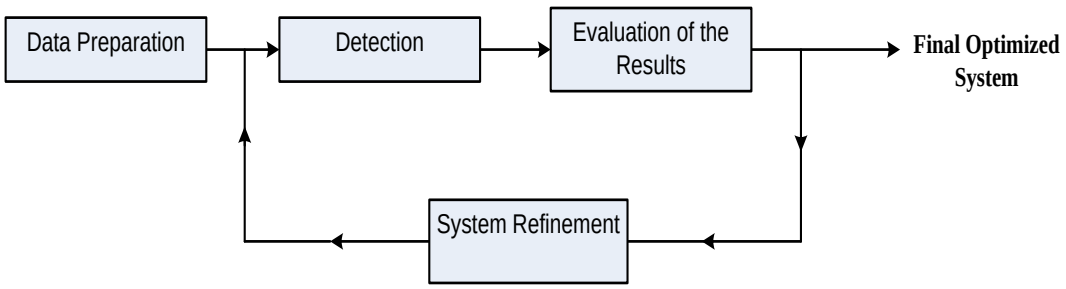


Figure (1): The stages of the Detection system

The general idea of the system refinement stage is to gradually enhance the performance of the detection through finding the set of optimum parameter values. This is done by inspecting the performance while varying some input parameter. After studying the effects of all parameters on the performance, we can achieve a sequence of certain steps that can be applied on any input image containing copy-move forgery. The set of optimized parameters may vary from image to image, but the algorithms and the order of parameters variation steps will be the same. The general form of the detection algorithm of copy-move forgery is showed in Figure (2). Different experiments are carried out and their results are presented in section 4.

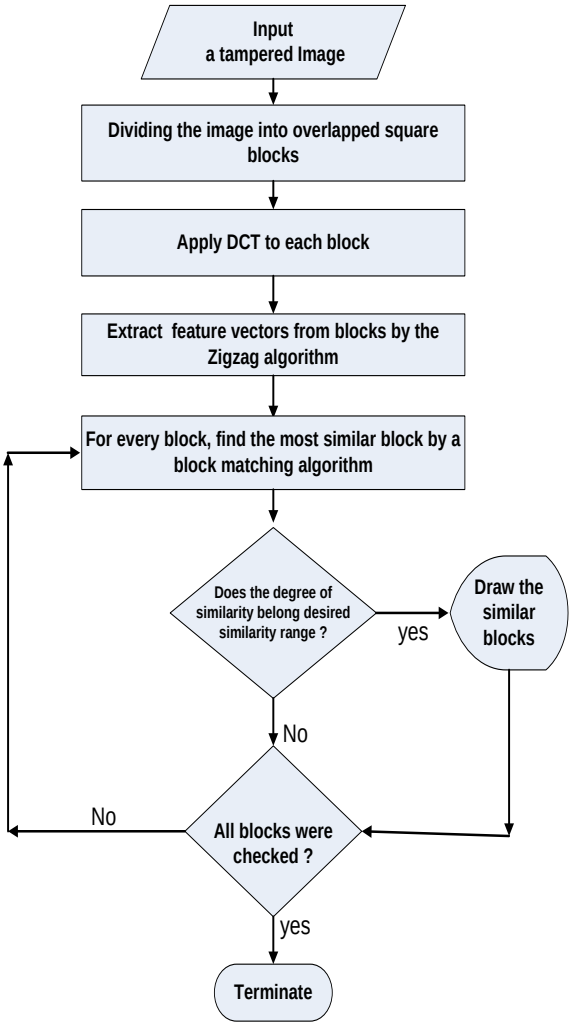


Figure (2): The copy-move detection algorithm

2.1 Features Extraction

A square block with a size B×B pixels slides along the image from the upper left corner right down to the lower right corner. The sliding will generate (M-B+1) (N-B+1) overlapped blocks.

The two-dimensional discrete cosine transform (DCT) is applied to every overlapped block. It is equivalent to a one-dimensional DCT performed along a single dimension followed by a one-dimensional DCT in the other dimension. The definition of the two-dimensional DCT [13] for an input image A and output image B is given by

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N} \qquad \textbf{(1)}$$

where $0 \le p \le M, \quad 0 \le q \le M,$

$$\alpha_p = \begin{Bmatrix} 1/\sqrt{M} & , & p = 0 \\ \sqrt{2/M} & , & 1 \le p \le M-1 \end{Bmatrix} ,$$

$$\alpha_q = \begin{Bmatrix} 1/\sqrt{N} & , & q = 0 \\ \sqrt{2/N} & , & 1 \le q \le N-1 \end{Bmatrix} ,$$

M and N are the row and column size of A, respectively.

Every block is represented by a feature vector. Basically, each feature vector is a row of a dimension B^2. In case of applying the DCT, the energy is focused on the first several values of transformed coefficients (the lower frequency coefficients). Therefore, the higher frequency coefficients can be truncated. The truncation process selects only a specified number of feature vector attributes by applying the zigzag method [9], as shown in Figure (3).

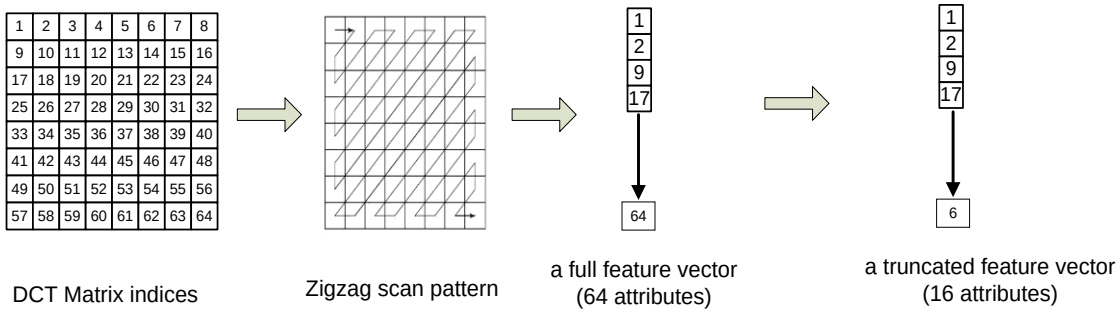


Figure (3): The Zigzag method to extract the most significant 16 coefficients

2.2 Matching step

As a first approach, the simple exhaustive searching technique is used for implementing the matching of feature vectors representing the overlapping blocks. This method computes the similarity distance from each block to all other blocks. This approach is very inefficient and its computational time complexity is of the order $O(N)$ [14]. To improve the efficiency of searching for neighboring feature vectors, some hierarchical structures have been proposed. The KD-tree search and lexicographical sort are commonly used algorithms for searching for nearest neighbors.

The degree of similarity of two blocks, B_i and B_j, can be measured by the Euclidean distance between the corresponding feature vectors. The similarity measure can be given as the following formula:

$$similarity \quad measure \quad S = \left(\sum_{k=1}^{\dim} \left(B_i[k] - B_j[k] \right) \right)^{1/2} \qquad \textbf{(2)}$$

The condition to accept two blocks to be similar is that $S = S_{\max}$, where the value $S_{\max}$ is of similarity distance that has the maximum number of occurrence. This is valid for tampered regions without added Gaussian noise or blurring.

A commonly used tool to conceal traces of tampering is the addition of white Gaussian random noise or blurring to the forged image regions. It is found that the performance of the detection is drastically decreased in the case of adding Gaussian noise and blurring [15].

A proposed solution of this problem is to change the previous condition into the following condition:

$$S_{\max} - E \leq S \leq S_{\max} + E \qquad \textbf{(3)}$$

Where E is the error in $S_{\max}$. In other words, there is an interval (or a range) of values that S can take to decide whether the two blocks are similar.

Another proposed method to enhance the performance of detection is to introduce another parameter D, whose value could be varied in order to decrease the false-negative value. D is defined as a threshold value of image distance (in pixels). If the distance between two given blocks d is below that threshold, then the two blocks are not similar. The image distance (in pixels) between two given blocks d is calculated from

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \qquad \textbf{(4)}$$

Where (x_1, y_1) and (x_2, y_2) are the coordinates of the upper-left pixel in first and second block, respectively. The following condition is applied:

IF $d > D$
THEN *the two blocks are similar and display them on the output screen***.**

To exclude pairs of blocks that have a large image distance between them, another parameter W is proposed. The value of W defines the range of image distances that separate the similar blocks. In other words, the following condition is applied:

IF $d > D$ *AND* $d < D + W$
THEN *the two blocks are similar and display them on the output screen***.**

In other words, a necessary condition to decide that two blocks are matched is

$$D < d < D + W \qquad \textbf{(5)}$$

Let us now divide the set of all overlapped blocks of the input image into two classes; C_1 , C_2. The first class C_1 contains blocks whose corresponding feature vectors satisfy the conditions (3) and (5) for a given user-defined values of the parameters E, D, and W. Those blocks will be painted on the output resulted image of the detection stage. The second class C_2 contains the rest of blocks.

2.3 Performance Evaluation of the system

The performance of the developed detection system can be evaluated by calculating both the accuracy of the detection and the false-negative. They are calculated from the following formulas [9]:

$$accuracy = \left[\frac{|T_1 \cap D_1| + |T_2 \cap D_2|}{|T_1| + |T_2|} \right] \times 100 \ \% \quad \textbf{(6)}$$

$$false-negative = \left[\frac{|T_1 \cup D_1| + |T_2 \cup D_2|}{|T_1| + |T_2|} - accuracy \right] \quad \textbf{(7)}$$

Where $|\ .\ |$ means the area of the region, $\cap$ means the intersection of two regions. The duplicated regions in a tampered image are denoted as T_1 *and* T_2, while the duplicated regions of an image detected by using our method is are denoted as D_1 *and* D_2.

In case of there are more than one duplicated regions in the tampered image, say n tampered regions, the accuracy of the detection and the false negative are calculated from the following formulas

$$accuracy = \left[\frac{|T_1 \cap D_1| + |T_2 \cap D_2| + ... + |T_{n+1} \cap D_{n+1}|}{|T_1| + |T_2| + ... + |T_{n+1}|} \right] \times 100 \ \% \quad \textbf{(8)}$$

$$false \ \ negative = \left[\frac{|T_1 \cup D_1| + |T_2 \cup D_2| + ... + |T_{n+1} \cup D_{n+1}|}{|T_1| + |T_2| + ... + |T_{n+1}|} - accuracy \right] \quad \textbf{(9)}$$

3. Experimental Results

In this section, the results of various detection experiments are presented. The performance of the developed forgery detection system is measured and the effects of variation of certain system's parameters are investigated. The accuracy of detection is computed for forged images with different shapes of tampered regions. The proposed algorithms are also tested in the case of horizontally and vertically flipped tampered regions. Also, experiments that detect tampered regions-with-holes are presented. The effects of adding Gaussian noise and blurring to the tampered images are studied. Three different methods of searching for similar blocks are compared. Different parameters of the developed

algorithms are varied in search of the optimum values, which lead to the highest accuracy of detection and the lowest false-negative values.

In the case of a block size of 8×8 pixels, the initial extracted feature vector is 64. After applying the zigzag algorithm, the typical size of the used truncated feature vectors is 16.

For all experiments in this section, a baseline dataset of gray-scaled images is used to generate different forged image datasets. It contains 50 natural images, which are selected from the data set [16]. Each image has the size 128×128 pixels. In all experiments, the average accuracy, average false-negative value, and average detection time are calculated from the results of the whole image dataset.

The input image is a gray image I of the size M×N. If it is a color image, it is converted to a grayscale image using the following standard formula [9].

$$I = 0.299R + 0.587G + 0.114B \qquad \textbf{(10)}$$

All the algorithms are developed with MATLAB 7. Experiments are worked out on a PC computer with 3.0 GHz Intel Core-2-Duo processor and 3.0 GB RAM.
In the beginning, a comparison between the performances of a three matching algorithms is made, as shown in Table (1).

Table (1): A comparison between different matching methods

Matching method	Time	Accuracy	false negative
Exhaustive search	720 sec	100 %	0
Lexicographical sort	58 sec	100 %	0
Kd-Tree search	61 sec	100 %	0

The results showed the superiority of the kd-tree search and lexicographical sort algorithms for the matching step. The time duration required to detect the tampered region is drastically reduced, compared to the exhaustive search.

The size of the *squared tampered region* is varied to test its effect on the accuracy of detection, as shown in Figure (4). The block size used in this experiment is 8×8 pixels. The results are compared with other work [10].

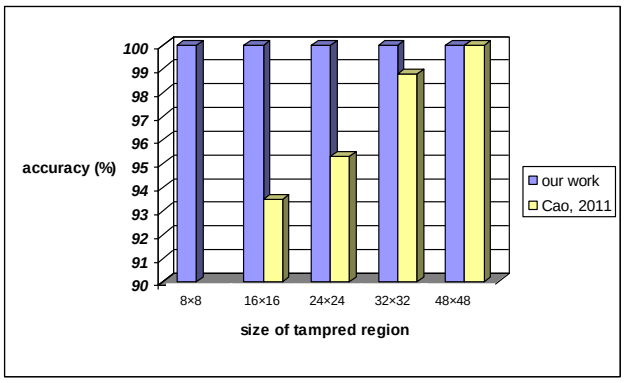


Figure (4): Variation of accuracy with different size of tampered regions

The results showed that the size of the squared tampered region does not affect the accuracy, given that squared tampered region is greater than or equal the block size.

(a) original image (b) an image with a tampered random region (c) detection result

Figure (5): Copy-move operation of random-shaped tampered region

The forged image may contain random tampered regions, as in Figure (5b). In this case, the results of detection yield that the accuracies are lower than results of the case of squared tampered regions as it is shown in Table (2). For each random region, the average accuracy of detection is calculated over all images in the image dataset. It is found that the accuracy depends on the shape of the tampered region. Some examples of random-shaped regions that are used in this work are shown in Figure (6).

(a) (b) (c) (d) (e)

Figure (6): Variation of accuracy of detection for different shapes

Table (2): Comparison of performance for different shapes

Shape of tampered region	Accuracy	False negative
squared	100 %	0
random-shaped (a)	99.6 %	0
random-shaped (b)	98.3 %	0
random-shaped (c)	89.4 %	0
random-shaped (d)	79.8 %	0
random-shaped (e)	58.2 %	0

For images with random-shaped tampered regions, in order to interpret the variations in the accuracies of detection, the following necessary condition of detection of tampered regions is stated:

"In a given tampered region, a pixel would be detected only if it is contained in a block that belongs to the class C_1*".*

There are two results of this condition; First, some of the fine details on the boundaries of the random regions can not be detected since some bumps (small emerging parts) on the boundaries that are smaller than the block size will not be detected (hence the accuracy decreases). Second, a whole random region can not be detected if its boundary does not include a squared area equals a block size.

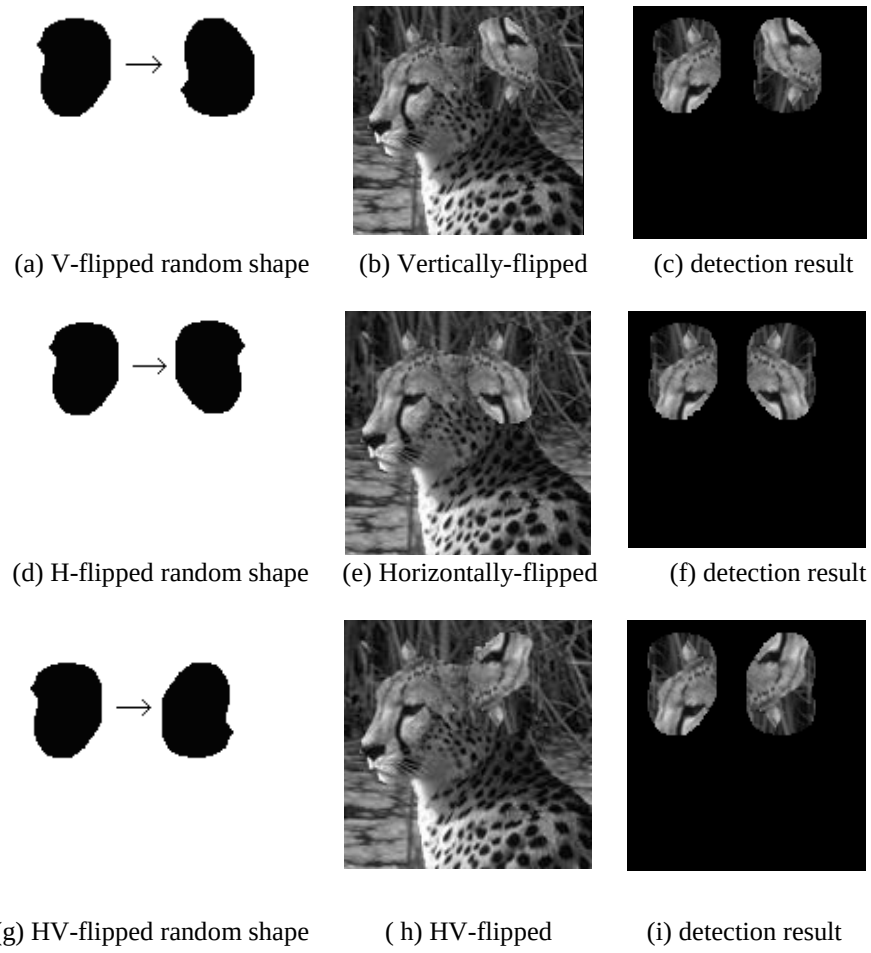
For images containing random-shaped tampered regions, the performance of the system is studied with the variation of the block size. The accuracies are decreased with the increasing of the block sizes. According to results shown in Table (3) for the random shape (a) in Figure (6), the best accuracies of detection are obtained with block sizes 8×8 and 6×6. The lower time duration of detection for the case of 8×8 make it the optimum choice. In general, the accuracies of detection of random-shaped regions depend on the shape of the random tampered region.

The experimental results showed high performance, not only for detecting squared tampered regions, but also for the detection of random-shaped tampered regions.

Table (3): Variation of block sizes for the case of random tampered regions

Block size	Time	Accuracy	False negative
6×6	76 sec	99.6 %	0
8×8	63 sec	99.6 %	0
10×10	48 sec	99.1 %	0
12×12	42 sec	97.8 %	0
14×14	36 sec	97.1 %	0
16×16	33 sec	95.5 %	0

The performance of detection is evaluated for the cases of vertically-flipped (VF), horizontally-flipped (HF), and VH-flipped tampered regions. As showed in Figure (7).



(a) V-flipped random shape (b) Vertically-flipped (c) detection result

(d) H-flipped random shape (e) Horizontally-flipped (f) detection result

(g) HV-flipped random shape (h) HV-flipped (i) detection result

Figure (7): Copy-move operation of flipped random-shaped tampered region

The results showed in Table (4) that the accuracies and false-negative values are not changed in case of flipping. The time durations needed for detection of the VF images or HF images are the same, but it is greater than the case of normal orientation. The case of both HF and VF needed more time to be detected. That is due to increasing of the number of feature vectors, and consequently the size of the kd-tree search.

Table (4): Results of Detection of flipped tampered regions

Type of flipping	Time	Accuracy	False-negative
Vertically random-shaped region	174 sec	99.6 %	0
Horizontally random-shaped region	174 sec	99.6 %	0
Vertically and Horizontally random-shaped region	540 sec	99.6 %	0

Another case is tested, as shown in Figure (8), in which the tampered region is a random region containing a hole. The results showed that the existence of a hole does not affect the accuracy of the detection, unless there are some of the small bumps on the outer boundaries or inner boundaries that are smaller than the block size

(a) original image (b)a random shape with a hole (c) detection result

Figure (8): Detection of a random tampered region containing a hole

The effects of adding either Gaussian blurring or additive white Gaussian noise (AWGN) to the tampered region are studied. In case of adding Gaussian blurring or noise to the images and the condition of the range of the similarity distance E = 0, the accuracy of detection is nearly 0 %. Therefore, the developed algorithm is changed to search for the most similar feature vectors (blocks) within a range of similarity distances as is described in sub-section3.2. First, the effects of variation of the parameters E, D, and W on the detection of tampered regions with added Gaussian noise are studied. For the optimum values of the parameters and by changing the signal-to-noise ratio, the performance of our system is compared to other research works [10], as in Table (5).

Table (5): Accuracy values for different noise degrees

S/N	Accuracy	
	Our results	**[Cao, 2011]**
25	95.8 %	95 %
30	99.8 %	97 %
35	99.9 %	99 %
40	100 %	100 %

The accuracies of detection are also studied in case of adding blurring to the tampered regions. For the optimum values of the parameters and by changing the blurring parameters, the performance of our system is compared to other research works [9, 10] in Table (6).

Table (6): Accuracy values for different blurring degrees

Filter Size w	σ	Accuracy		
		our results	[Huang, 2011]	[Cao, 2011]
3×3	0.5	100 %	92.2 %	92 %
3×3	1	97.7 %	91.5 %	91.2 %
5×5	0.5	98 %	88 %	89 %
5×5	1	91.9 %	86 %	90 %
7×7	0.5	95.2 %	-	-
7×7	1	89.19%	-	-

The Gaussian noise can be added either on the tampered regions or on the whole image. Comparisons are made between the performances of the system for these two cases. Moreover, these comparisons are made for different sizes of tampered regions.

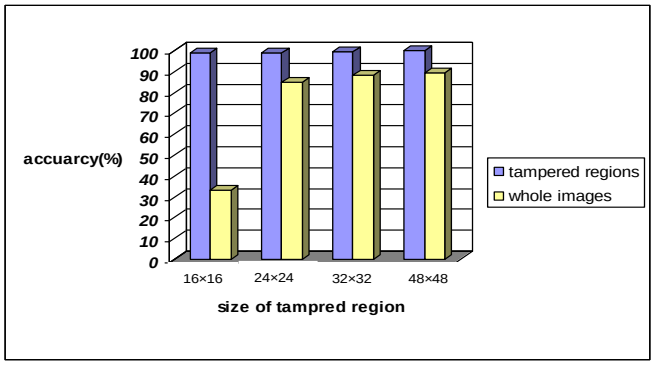


Figure (9): Comparison between the effects of adding Gaussian noise to the whole images and only on tampered regions

It is found that the accuracy of detection is better in case of adding noise to just tampered regions, compared to the case of adding noise to the whole image, as shown in Figure (9). On the contrary, the two cases yield almost the same accuracies in case of adding blurring.

In case of added noise or blurring, using the kd-tree algorithm resulted in better performances compared to using the lexicographical sort, as shown in Figure (10).

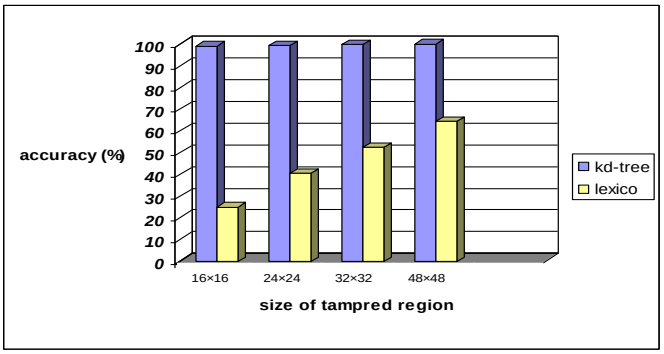


Figure (10): Comparison between two matching algorithms (With added noise on tampered regions)

The Gaussian noise and blurring are added to the random-shaped regions.

Table (7): Different noise degrees on random regions

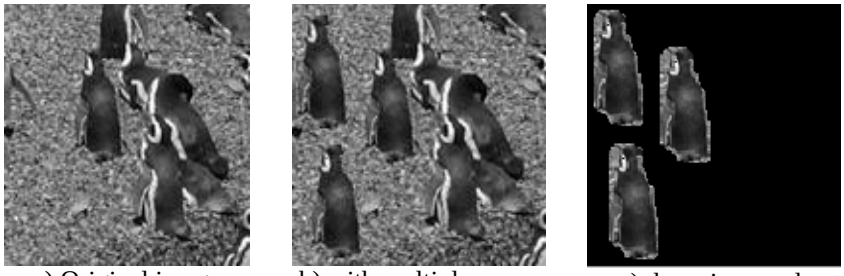
SNR/db	Accuracy	False-negative
25	93 %	0.137
30	94.1 %	0.020
35	97.3 %	0.031
40	99.1 %	0.037

Table (8): Different blurring degrees on random regions

Filter Size w	σ	Accuracy	false-negative
3×3	0.5	99.6%	0.082
3×3	1	98.0%	0.134
5×5	0.5	99.6%	0.113
5×5	1	86.8%	0.110

From Table (7) and Table (8), the accuracies decreased with increasing the degrees of added noise by degrees (S/N) and blurring. However, the accuracies are acceptable in case of adding noise or blurring.

In case of multiple tampered regions, as shown in Figure (11), our detection algorithm is also able to detect all tampered regions.



a) Original image b) with multiple copy-paste c) detection result

Figure (11): Detection of multiple tampered regions

In case of added Gaussian noise and blurring, the system's parameters should be manually varied to gradually enhance the performance of the detection. The best practical steps to reach the optimum performance will be summarized as in the following.

In our implemented detection algorithm, we included three parameters; E, D, and W. They are defined in section 2. They should be adjusted according to each input tampered image. At the beginning of experiments, their initial values are E = 0, D = 1, and W= 100.

While inspecting the accuracy of detection and the false-negative values, the value of E should be gradually increased. This process is continued as long as the accuracy of the detection is increasing. When the accuracy stops increasing, then the value of D should be increased. As a result of this step, the value of the false-negative is decreased. At a certain threshold value of D, the copied and the moved regions will disappear. Therefore, the value of D should be decreased just below the threshold value to uncover the two regions. Then, the value of W should be manually decreased. As a result, the value of the false-negative is decreased. This process is continued until the accuracy of the detection is affected.

After applying the above methods, the accuracy and false-negative values are optimum for the given input tampered image.

4. Conclusions

Detection of copy-move forgery in images is a very active research field in image forensics. In this paper, a copy-move forgery detection algorithm is proposed. It is based on applying DCT followed by the zigzag method to extract feature vectors and the kd-tree algorithm as a block-matching method. Our algorithm can efficiently detect both squared and random-shaped tampered regions. It can also detect vertically and/or horizontally flipped tampered regions. Moreover, it can detect tampered region containing holes, intersected regions, and multiple copy-move regions. The experimental results showed that the accuracy of detection of non-distorted squared tampered regions is 100 %. In case of random-shaped tampered regions, the results yield that the accuracies are lower than the case of squared tampered regions. It is found that the accuracy depends on the shape of the tampered region.

In case of distorted tampered regions, different parameters are varied to find their optimum values that yield the highest accuracies of detection. The accuracy of detection reached 95.8 % for the case of adding a white Gaussian noise (S/N = 25), and 91.9 % for applied blurring (σ =1, w=5×5).

Experimental results showed that our proposed methodology, including methods and parameters, is robust and efficient for different tampering cases. This methodology can be automated by implementing it through a computer program.

References

[1] B. Mahdian and S. Saic; “A bibliography on blind methods for identifying image forgery”.
Signal Processing: Image Communication, vol. 25, pp.389-399, 2010.

[2] M. Schneider and S. Chang; “A robust content based digital signature for image authentication”. *IEEE International Conference on Image Processing* (ICIP’96), vol.3, pp. 227-230, 1996.

[3] H. Sencar, M. Ramkumar and A. Akansu; “Data Hiding Fundamentals and Applications:

Content Security in Digital Multimedia". *Academic Press, Inc., Orlando, FL, USA,* 2004.

[4] M. Arnold, M. Schmucker and S. Wolthusen; "Techniques and Applications of Digital Watermarking and Content Protection". *Artech House, Inc., Norwood, MA, USA*, 2003.

[5] J. Fridrich, D. Soukal and J. Lukas; "Detection of copy-move forgery in digital images".*Proc Digital Forensic Workshop,* pp. 19-23, 2003.

[6] A. Popescu and H. Farid; "Exposing Digital Forgeries by Detecting Duplicated Image Regions". *Dartmouth College, Tech. Rep*. TR2004-515, 2004.

[7] F. Shih and A. Yuan; "A Comparison Study on Copy-Cover Image Forgery Detection". *The Open Artificial Intelligence Journal*, Vol. 4, pp. 49-54, 2010.

[8] S. Kumar, S. Mukherjee and P. Das; "Copy-Move Forgery Detection in Digital Images: Progress and Challenges". *International Journal on Computer science and Engineering(IJCSE)* Vol. 3, issue 2, pp. 652-663, 2011.

[9] Y. Huang, W. Lu, W. Sun, and D. Long; "Improved DCT-based detection of copy-move forgery in images". *Forensic Science International.* Vol. 206, pp. 178–184, 2011.

[10] Y. Cao, T. Gao L. Fan and Q. Yang; "A Robust Detection Algorithm for Region Duplication in Digital Images". *International Journal of Digital Content Technology and its Applications.* Vol. 5, Number 6, pp. 95-103, 2011.

[11] W. Luo, J. Huang and G. Qiu; "Robust detection of region-duplication forgery in digital image". *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06*), pp. 746-749, 2006.

[12] G. Liu, J. Wanga, S. Lian and Z. Wanga; "A passive image authentication scheme for detecting region-duplication forgery with rotation". *Journal of Network and Computer Applications,* vol.34, issue 5, pp. 1557-1565, 2011.

[13] A. Jain; "Fundamentals of Digital Image Processing". Englewood Cliffs, NJ, Prentice Hall, 1989.

[14] B. Mahdian and S. Saic; "Detection of copy–move forgery using a method based on blur moment invariants", *Forensic Science International*, vol. 171, pp. 180-189, 2007.

[15] B. Mahdian and S. Saic; "Using noise inconsistencies for blind image forensics". *Image and Vision Computing,* vol. 27, pp. 1497-15, 2009.

[16] T. Ng, J. Hsu and S. Chang; Columbia Image Splicing Detection Evaluation Dataset, http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/ AuthSplicedDataSet.htm/. (Last accessed: May, 2012).