Enhanced Filter-based SIFT Approach for Copy-Move Forgery Detection

Mohamed A. Elaskily*, Heba K. Aslan*, Mohamed M. Dessouky, Fathi E. Abd El-Samie***, Osama S. Faragallah****, Osama A. Elshakankiry******

* Dept. of Informatics, Electronics Research Institute.

** Dept. of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University.

*** Dept. of Electronics and Electrical Communications, Faculty of Electronic Engineering, Menoufia University.

**** Dept. of Information Technology, College of Computers and Information Technology, Taif University

(Received: 21 Mar. 2018 – Accepted: 1 Apr. 2018)

Abstract

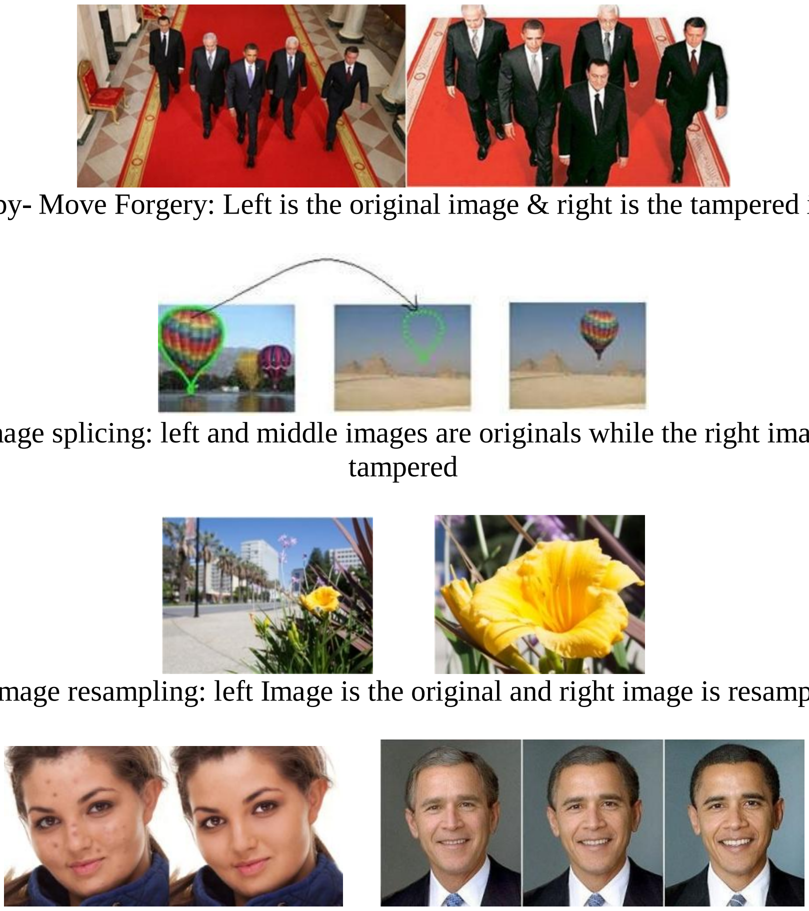
Image forgeries are applied to give the digital images other meanings or to deceive the viewers. Image forgeries appear in many cases such as judges in courts, cybercrimes, military and intelligence deception, or defamation of important characters. There are many different types of image forgeries such as copy move forgery, image retouching, image splicing, image morphing, and image resampling. Copy move forgery is the widest type and easy to apply between all digital image forgeries. Scale Invariant Features Transform (SIFT) algorithm is used strongly to detect copy move forgeries due to its efficiency in digital image analysis. SIFT algorithm is extracting image features, which are invariant to geometrical transformations such as scaling, translation, and rotation. These features are used in performing the matching between different views of a scene or an object. This paper enhances the efficiency of using SIFT algorithm in detecting copy move forgery by two ways. Firstly, it enhances the image itself by applying different types of digital filters to reinforce the image features giving the ability to detect forgeries. Butterworth low-pass filter, a high-pass filter, and the combination of them are applied to this task. Secondly, the matching strategy is adapted based on a new thresholding approach to increase the true positive rate and decrease the false positive rate. Experimental results show that the proposed approach gives better results compared with traditional copy-move detection approaches. In addition, it gives better stability and reliability to different copy-move forgery conditions.

1. Introduction

There are two types of authentication; active and passive authentication [1]. Active authentication is applied by using a digital signature or digital watermarking. It depends on using the original content of the digital image before any use of this image [2]. On the other hand, passive authentication is used in image forensics to trust an image without any previous knowledge about the image contents [3]. Digital image forgery has several types such as copy move forgery, image splicing forgery, image resampling forgery, image retouching forgery, and image morphing forgery. Copy move forgery duplicates some parts of an image and hides some details [4]. Image splicing forgery uses two or more images to compose another image consisting of a collection of different objects from each image [5]. Image resampling creates a new image with increasing/decreasing height/width of a specific object [6]. Image retouching forgery enhances some parts of the image to reveal or hide some features [6]. Image morphing forgery creates a new object from two different objects in different images. [4]. Figure 1 shows examples of all the above-mentioned forgery types.

There are several algorithms that are used to detect copy move forgeries such as Discrete Cosine Transform (DCT) [7], Discrete Wavelet Transform (DWT) [8], Principal Component Analysis (PCA) [9], and Singular Value Decomposition (SVD) [10]. These algorithms extract coefficients from the transform domains of the image. They use these extracted coefficients as features arranged in a lexicographical map to search for the duplicated features [7]. These algorithms are block-based algorithms as they divide the image into small blocks.

There are other algorithms, which use different types of features that describe the image objects and their orientations and intensity levels to detect rotation, scaling, and translation. These algorithms are based on the notion of invariant image moments [11]. Other algorithms are texture and intensity descriptor algorithms, which use the structure of the digital image obtained from statistical features appearing frequently in different patterns of the image. The relationships between patterns properties can be utilized to detect the tampering by discovering distortion in the texture patterns of an image and texture descriptors [12].



a) Copy- Move Forgery: Left is the original image & right is the tampered image

b) Image splicing: left and middle images are originals while the right image is tampered

c) Image resampling: left Image is the original and right image is resampled

d) Examples of image retouching e) Examples of image morphing

Fig. 1: Digital image forgery shapes

The most efficient algorithms to defeat copy move forgery are invariant key-points algorithms that depend on a large number of local features extracted from an image. These features are characterized by stability against rotation, scaling, translation and partial stability against illumination changing. Algorithms based on this trend are classified as non-block based algorithms as they extract features from the whole image [13].

The rest of the paper is organized as follows. The next section shows the related works. In section 3, the proposed approach is discussed in details. In section 4, experimental results on different datasets are presented with different types of attacks. Finally, Section 5 concludes the paper.

2. Related Work

Invariant keypoints based algorithms extract features from a whole image. Speed Up Robust Features (SURF) and Self Invariant Features Transform (SIFT) algorithms are the most suitable image features descriptors used to extract local features from an image [14]. These features are invariant against translation, scaling, rotation, and sometimes discover changes in contrast and brightness (illumination) [15]. SIFT algorithm passes through four main steps [16] that are illustrated in next steps and shown in Figure 2:

a) Detection of scale space extrema: for an image $I(x\ .y)$, the function of Gaussian $G(x,\ y,\sigma)$ is used to get a scale space kernel $L(x,\ y,\sigma)$ as shown in equation 1.

$$L(x,\ y,\sigma) = G(x,\ y,\sigma) * I(x,\ y) \tag{1}$$

where * is the function of monitoring how the values of x and y are modified in each scale space conversion window. To find the most keypoint locations stability, the scale space extrema determined from Difference-Of- Gaussian (DOG) is applied as shown in equation 2 and equation 3.

$$D(x,\ y,\sigma) = (G(x,\ y,k\sigma - G(x,\ y,k\sigma) * I(x,\ y) \tag{2}$$

$$[D(x,y,\sigma) = L(x,y,k\sigma\sigma) - L(x,y,\sigma)] \tag{3}$$

where k is a constant multiplicative factor that represents number of separated scales and $G(x,\ y,k\sigma)$ is a Gaussian blur function.

b) Keypoint localization: After determination of candidate keypoints, a detailed data about each one contains location, scale, and ratio of dropping. This data reflects keypoint stability, which is used to select the best keypoints mostly stable in location and scale.

c) Orientation assignment: In this step, each keypoint has a constant orientation based on local image gradient directions.

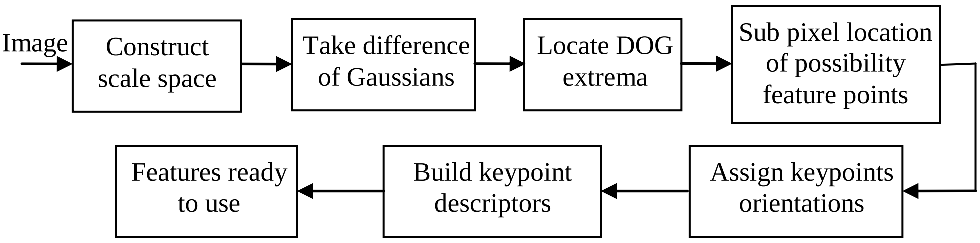


Image
Construct scale space
Take difference of Gaussians
Locate DOG extrema
Sub pixel location of possibility feature points
Assign keypoints orientations
Build keypoint descriptors
Features ready to use

Fig. 2: Steps of SIFT algorithm

The keypoint orientation is calculated from an orientation histogram of local gradients from smoothed image $L(x, y, \sigma)$ for each image sample $L(x, y)$ at the keypoint scale σ. The gradient magnitude $m(x, y)$ and orientation $\theta(x, y)$ are computed using pixel differences as shown in equation 4 and equation 5.

$$m(x, y) = \sqrt{L_1^2 + L_2^2} \qquad (4)$$

$$\theta(x, y) = \arctan(L_2 / L_1) \qquad (5)$$

d) Keypoint descriptor representation: A keypoint descriptor is created by computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location as shown in Figure 3. Each descriptor consists of 128 elements as a 4x4 window sub-patches descriptor width with 8 magnitudes $m(x, y)$ and orientation angles $\theta(x, y)$. So, the final SIFT feature vector has 128 elements.

The first usage of SIFT algorithm in copy move forgery detection appeared in [17], which simply extracts SIFT descriptors of an image. The similarity between duplicated regions results from a copy/paste operation. The algorithm is robust against image processing attacks such as noise addition, JPEG compression and robust also against rotation. Unfortunately, it is not efficient with scaling and translation.

Another algorithm uses SIFT features to detect copy move forgery is in [18]. Firstly, it finds image keypoints and collects features using SIFT algorithm. Secondly, by using the best-bin-first algorithm [19], keypoints matching is performed, and then the algorithm estimates the geometric distortions with affine transform by employing Random Sample

Consensus (RANSAC). This algorithm shows robustness against geometric distortions, rotation, scaling, and illumination changes.

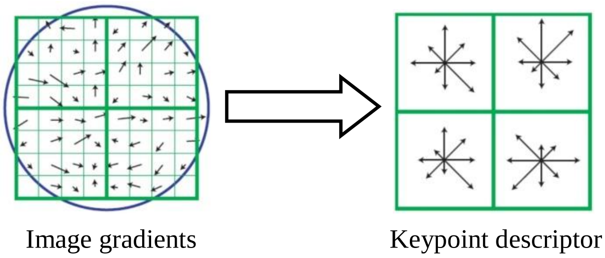


Fig. 3: Keypoint descriptor representation

Other algorithms use a hybrid approach from one or more techniques with SIFT algorithm such as the algorithm in [20]. This algorithm applies DWT to use low frequency sub-band (LL) from the transformed image containing most image information for image reduction and then SIFT algorithm applies on this sub-band. This reduction aims to increase accuracy and decrease complexity.

The algorithm in [21] applies SIFT algorithm first to extract SIFT features and then applies Singular Value Decomposition (SVD) method for matching between these features and detecting duplicated regions.

One of the most powerful algorithms in copy move forgery detection and transformation recovery is presented in [22]. It uses SIFT algorithm to detect geometric transformations like scaling, rotation, translation and gives good results against image processing attacks like Gamma correlation, JPEG compression, and signal to noise ratio. However, it is vulnerable to copy move attacks existing in full resolution images and it requires large complexity as well as large computational time.

3. Proposed Method

The proposal is based on SIFT algorithm to generate SIFT features of an image and search for matched features resulting from duplicated regions in the digital image. These features may be slightly close, especially in small patches forgery, which causes misleading results. There are many different types of image processing attacks such as image blurring, noise addition, JPEG compression, brightness changing, and contrast adjustment in addition to the traditional attacks such as scaling, rotation, and translation. All these attacks hide the copy move forgery. It is clear that the increase in image smoothing, noise reduction and image sharpening are needed to exhibit details of edges to discover forgeries existing in edge areas. Also, copy move attack may cause edges in strange locations, because it copies a patch from a location in the image with brightness degree and moves it in another location with a different brightness degree. The proposal achieves these targets by applying the following steps that are shown in Figure 4:

a) Converting the color image into a gray-scale image to reduce computation complexity.
b) Applying digital image filtering using a high-pass filter, Butterworth low-pass filter, or combination of them.
c) SIFT features are extracting from the pre-processed image, and mapping these features for matching.
d) Building hierarchal clustering with a new strategy.
e) Detecting forgery and estimating geometric operations.

4.2 Gray-Scale Conversion and Image Filtering

In this step, two processes are performed. Firstly, the tested color image is converted into a gray-scale image to reduce the image size without loss of features. Secondly, one filter or a combination of digital filters is applied to enhance the image features to be accurate and easy to extract from the image. The utilization of these filters overcomes any fake matching caused by similarity between two different features, and faces image processing attacks like noise adding, image blurring, JPEG compression, and brightness change.

1) High pass filter (HPF) is used to eliminate low frequencies, which is useful in enhancing the image sharpness and increasing the details of the edges. Using the HPF increases the features extracted from the image. Using an ideal HPF increases image sharpening and avoids the problem of image ringing effect. The proposal uses a high-pass filter with equation 6.

$$H(u\ ,v) = \begin{cases} 1 & if\ D(u,v) \le D_o \\ 0 & if\ D(u,v) > D_o \end{cases} \qquad (6)$$

where the input image $f(x, y)$ with size $(M \times N)$ is filtered by function $H(u,\ v)$, $u = (0,1,2,\ldots, P-1)$, $v = (0,1,2,\ldots, Q-1)$, $(P \times Q)$ is the size of padded image with padding parameters $P \ge 2M - 1$, $Q \ge 2N - 1$, D_0 is the cutoff frequency represented as a radius of a circle containing allowed frequencies and $D(u,v) = [(u - P/2)^2 + (v - Q/2)^2]^{1/2}$.

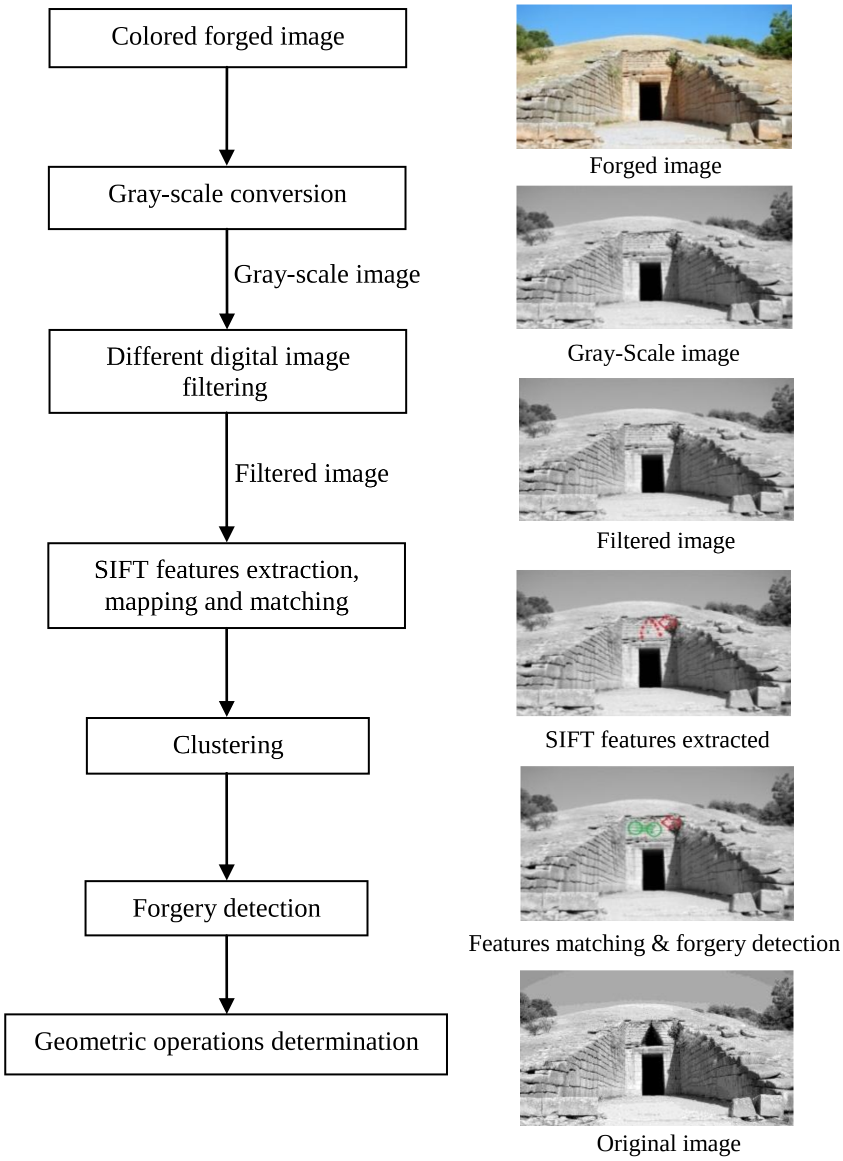


Fig. 4: Flowchart of proposal steps in addition to example with images in each step

2) Butterworth low pass filter (BLPF) is used to eliminate high frequencies. It is useful in noise reduction and image smoothing. BLPF

passes all frequencies within a circle of radius D_0 and attenuates all other frequencies outside this circle. The BLPF is a type of low pass filter which has order (n). The main goal of applying BLPF is to smooth the image and to reduce noise. Also, the BLPF overcomes the problem of ringing effect. Ringing effect appears as bands or ghosts near edges. BLPF is applied with equation 7.

$$H(u,v) = 1/1 + [D(u,v)/D_0]^{2n} \qquad (7)$$

where (n) is the order of butterworth low pass filter, D_0 is cutoff frequency represented as radius of a circle and $D(u,v) = [(u-P/2)^2 + (v-Q/2)^2]^{1/2}$.

4.2 Extracting SIFT Features, Mapping and Matching

An image x has a number of keypoints $x_1, x_2, \ldots\ldots, x_n$. Firstly, SIFT algorithm is used to extract SIFT descriptors $f_1, f_2, \ldots\ldots, f_{128}$, which meet each keypoint. Secondly, it applies similarity search between SIFT keypoints to detect primary similar keypoints among all SIFT area space. A similarity vector $v = \{v_1, v_2, \ldots\ldots, v_{n-1}\}$ is constructed, which represents the stored Euclidian distance between similar SIFT keypoints. Some features are naturally similar to others due to their same location. So, addition condition in matching operation is needed to distinguish between features. By training, a threshold T with value identified between 0.5 and 0.6 is used. Based on the rule, keypoints that are different show high values of Euclidean distance among them and keypoints that are similar show low values of Euclidean distance among them.

$$v_i / v_{i+1} < T \text{ where } T \in (0.5, 0.6) \qquad (8)$$

Equation 8 shows that if the ratio of distance between one keypoint and its neighbor v_i from a side and distance between same keypoint and second step neighbor v_{i+1} in the other side is larger than T, the features are random and no matching occurs. Otherwise, if the ratio is less than the threshold T the similarity match occurs. The overall hierarchy of that step is shown in Figure 5.

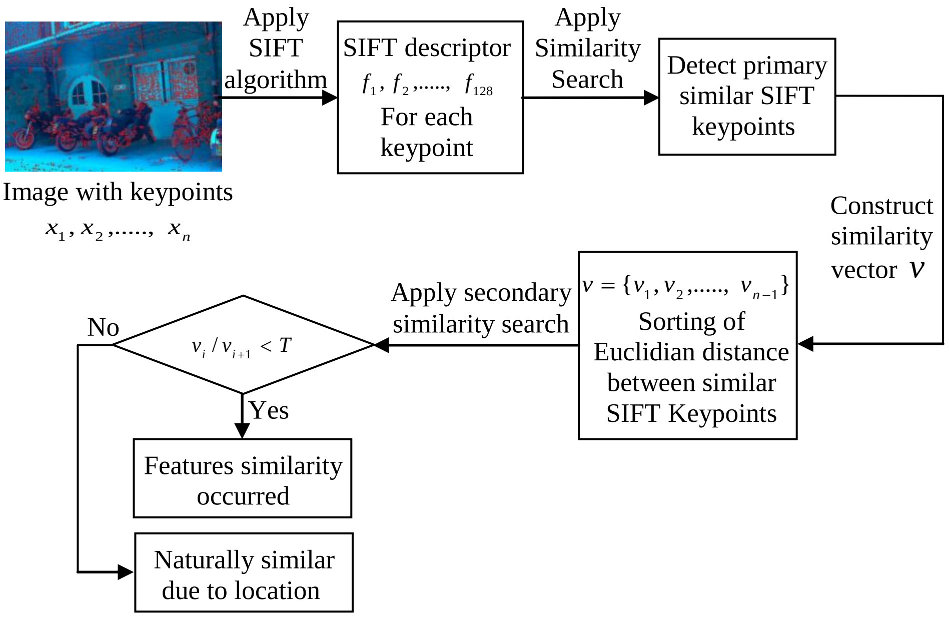

Apply SIFT algorithm
SIFT descriptor $f_1, f_2, \ldots\ldots, f_{128}$ For each keypoint
Apply Similarity Search
Detect primary similar SIFT keypoints
Image with keypoints $x_1, x_2, \ldots\ldots, x_n$
Construct similarity vector v
$v = \{v_1, v_2, \ldots\ldots, v_{n-1}\}$ Sorting of Euclidian distance between similar SIFT Keypoints
Apply secondary similarity search
$v_i / v_{i+1} < T$
No
Yes
Features similarity occurred
Naturally similar due to location

Fig. 5: Overall hierarchal of SIFT features extraction, mapping and matching.

4.2 SIFT Features Clustering

In this step, the extracted SIFT features in the previous step are used to determine similarity to detect cloned areas by finding matched features. Matched features with spatial locations that are close to each other are collected by applying a clustering operation. Each Keypoint is distributed in a cluster to construct a number of clusters equal to the number of keypoints. The mutual spatial distance between all clusters is calculated. A merging process is applied between clusters to merge the nearest clusters according to spatial linkage between them and a threshold is used to stop the clustering operation. There are many types of linkage methods to use such as single linkage, which depends on smallest Euclidian distance between any two objects or keypoints in two clusters.

Centroid linkage is a second merging method. It depends on the Euclidean distance between two objects or keypoints in the center of each cluster. Ward linkage is a third linkage method, which depends on the increase or decrease in Error Sum of Square (ESS) as illustrated below. Two clusters A and B consist of n_A and n_B objects or keypoints. x_{A_i} and x_{B_j} are the order of each object or keypoint in the clusters A and B, respectively. Ward linkage method is applied according to equation 9 and equation 10:

$$\Delta dist\ (A, B) = ESS\ (AB) - [ESS\ (A) + ESS\ (B)] \tag{9}$$

$$ESS\ (A) = \sum_{i=1}^{n_A} (|x_{A_i} - x_A|)^2 \tag{10}$$

where A and B are the combined clusters, x_A is the centered object or keypoint in cluster A , and ESS is the value of error resulting from dividing the image into equivalent squares. When the value of ESS=0, this reflects that all the keypoints and objects of the image are in one cluster. In addition, another threshold T_h is used to determine where to cut off the clustering process and reach the final number of clusters by comparing with Inconsistency Coefficient IC . The average distance between all clusters in the final hierarchy built is calculated using ESS. The higher values of IC reflect less similar keypoints. High values of IC are the second criterion to stop clusters merging operation. Also, high values of IC do not allow far patches to be merged according to spatial location.

4.2 Geometric Operations Determination

In the previous steps, the proposal examined if there are matching keypoints or objects in the image to detect if the image is forged or not. If the image is detected as tampered, the geometric operations applied to hide forgery are detected as follows:

1) By using at least five matched pairs of points, where a point is located by (x, y) and the matched point is located by (x^*, y^*), an affine homography function H is represented by equation 11.

$$(x^*, y^*, 1) = H\,(x, y, 1) \tag{11}$$

2) After calculating (H) function using the five pairs of points as a reference, all other points are transformed based on the calculated H.

3) By using an iterative process called Random Sample Consensus Algorithm (RANSAC) to calculate the distance between transformed points and their original points, a model from the set of observed data is built.

4) If the distance is lower than a threshold β, the point is identified as an inlier point, and if the distance is above the threshold β, the point is identified as an outlier point.

5) From the previous iterations, the matching procedure focuses on the inliers points only. The proposal calculates the composition matrix *A* referring to the geometric transformation done as in equation 12.

$$A = R(\Theta)(R(-\Phi)SR(\Phi)) \qquad (12)$$

where *A* is identified from the diagonal matrix H as in equation 13.

$$H = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \qquad (13)$$

We calculate the reverse function R by applying reverse rotation and reverse scaling. Reverse rotation is implemented with rotation angle Φ and rotation direction ϴ. The reverse scaling values are calculated by a diagonal matrix $S = diagonal\ (S_1, S_2)$ for scaling transformations applied on point (x, y) .

4. Experimental Results

In this section, the results focus on estimating the difference between the proposal results and the earlier methods results in copy move forgery detection. The proposal is tested using a machine, which has Intel core i7 64 bits processor with 8 GB RAM and works by Linux Ubuntu 14.04.3.

4.1 Datasets

This work uses four different datasets to show the results and evaluate its efficiency in different conditions compared with other methods. These datasets are MICC-F220 [22], MICC-F2000 [22], MICC-F600 [23] and SATS-130 [24]. MICC-F220 dataset contains 220 images with about 737 × 492 resolution, and it is divided into 110 original images and 110 tampered images. The patched regions used in copy move forgery are about 1.2% of the whole images. MICC-F2000 dataset consists of 2000 images with about 2048 × 1536 resolution, and it is divided into 1300 original images and 700 tampered images. The copy moved patched regions are about 1.12% of the whole images. These patched regions are exposed to attacks such as rotation, scaling, and translation with different values of angles ϴ and $S = (S_1, S_2)$ such as illustrated in [22].

MICC-F600 dataset is composed of 600 images with different resolutions varying from 800 × 533 to 3888 × 2592 pixels, and it is divided into 448 original images and 152 tampered images. MICC-F600 is divided into four parts; the first part contains 38 images with one duplicated region after translation. The second part contains 38 images with two or three duplicated regions after translation. The third part contains 38 images with one duplicated region after rotation by angle 30^{o}. The fourth part contains 38 images with one duplicated region after rotation by 30^{o} and scaling by 120%. MICC-F600 is a strong testing dataset, because it has very small differences between original images and forged images such as in Fig. 6. MICC-F600 is also exposed to other forgeries applied in copied patched regions before duplication like adding Gaussian noise and added JPEG artifacts.

Dataset SATS-130 is used in testing many methods like that in [24]. It consists of 96 images with variable resolutions between 1024 × 683 and 3264 × 2448, and it is divided into 48 original images and 48 forged images. MICC-F2000, MICC-F600, and SATS-130 contain forged images with multiple copied regions, where one or more regions were copied and pasted in several regions in the image.

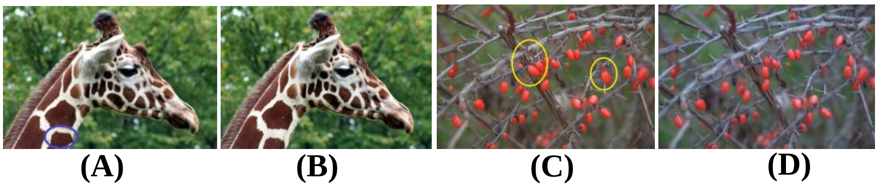


Fig. 6: Small differences between forged images and original images, where (A) and (C) are the forged images, while (B) and (D) are the original images

4.2 Testing Metrics

When testing an image, there are four metrics:

a) True Positive (*TP*) which is the number of tampered images that were correctly identified by the detection algorithm as tampered.
b) False Positive (*FP*) which is the number of original images that were incorrectly identified as forged images.

c) False Negative (*FN*) which is the number of forged images that were incorrectly identified as original.

d) True Negative (*TN*) which is the number of original images that were correctly identified as original. These parameters are summarizes in Fig. 7.

The above-mentioned quantities, performance evaluation were tested in terms of True Positive Rate (*TPR*), which is the probability that the forged image is detected. TPR is calculated as $TPR = TP/(TP + FN)$ and the remainder value is the False Negative Rate (*FNR*). *FNR* is calculated as $FN/(FN + TP)$ or $1 - TPR$. False Positive Rate (FPR) is the probability that the original images are not correctly identified. FPR is calculated as $FPR = FP/(FP + TN)$, and the remainder value is called True Negative Rate (*TNR*). *TNR* is calculated as $TN/(TN + FP)$ or $1 - FPR$.

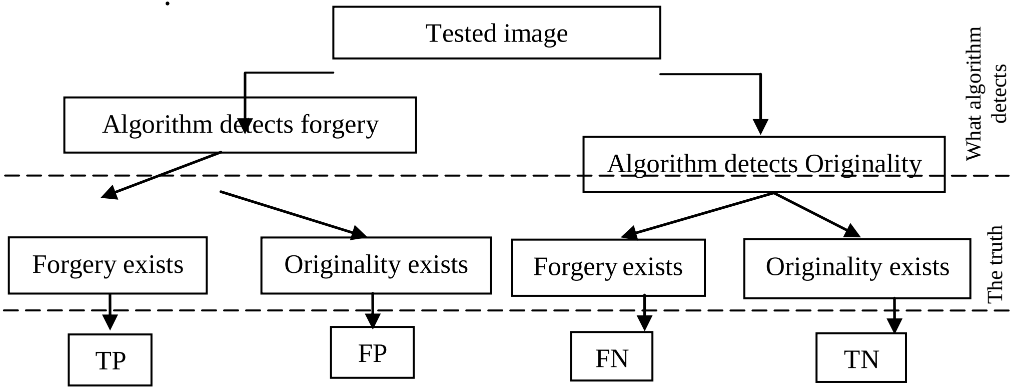


Fig. 7: Testing metrics.

4.3 Results with Different Datasets

In this section, three different types of filters are applied on the different datasets. We will compare between the proposed algorithm results and results of other methods such as Amerini et al. [22], Amerini et al. [23], and Christlein et al. [24], which used the same datasets. The first experiment starts by applying a high-pass filter (HPF) on different datasets. The algorithm is applied with a minimum number of points allowed to be similar and a threshold T_h for ward linkage method of 2. The results are listed in Table 1.

Table 1: Metric parameters values after applying high-pass filter and applying SIFT algorithm & forgery detection

	MICC-F220	MICC-F600	MICC-F2000	SATS-130
TPR %	99.09	89.24	95.1	76.2
FPR %	9.01	7.13	7.2	11.33
TNR %	90.99	92.87	92.8	88.67
FNR %	0.91	10.76	4.9	23.8

The second test is performed by applying a low-pass Gaussian filter (LPGF) on different datasets and continuing the algorithm steps with the same values of minimum matched points and T_h . This test uses different values of cutoff frequencies (*fc*) in LPGF from 160 to 220. The LPGF results are listed in Table 2.

Table 2: Metric parameters values after applying low-pass Gaussian filter and applying SIFT algorithm & forgery detection with variable values of cutoff frequency

	MICC-F220				**MICC-F600**			
	TPR %	**FPR %**	**TNR %**	**FNR %**	**TPR %**	**FPR %**	**TNR %**	**FNR %**
***fc*=160**	98.18	19.09	80.91	1.82	80.2	22.03	77.97	19.8
***fc*=180**	97.87	14.55	85.45	2.13	87.5	16.1	83.9	12.5
***fc*=200**	98.18	19.09	80.91	1.82	85.1	18.02	81.98	14.9
***fc*=220**	96.01	13.55	86.45	3.99	82.7	20.13	79.87	17.3
	MICC-F2000				**SATS-130**			
	TPR %	**FPR %**	**TNR %**	**FNR %**	**TPR %**	**FPR %**	**TNR %**	**FNR %**
***fc*=160**	89.3	17.6	82.4	10.7	71.73	16.83	83.17	28.27
***fc*=180**	94.8	12.1	87.9	5.2	79.32	27.51	8473.	20.68
***fc*=200**	91.2	16.1	83.9	8.8	74.8	14.2	85.8	25.11
***fc*=220**	87.2	21.01	78.99	12.8	72.3	11.75	88.25	27.7

From Table 2, it is noticed that the best result among the four datasets occurs at a cutoff frequency fc=180, except the TPR in MICC-F220. Over fc=200, the efficiency is decreased and the complexity is increased. The high complexity comes from the large number of extracted features; approximately from 8000 to 16000 features per image in average. These results are close to those in [22], [23] and [24], because LPGF has a large effect on noise reduction and image smoothing making the extracted SIFT features similar to each other, and causing large values of FPR as shown in Table 2.

The third test is carried out by applying Butterworth low-pass filter (BLPF) on the same datasets. The results of applying BLPF are listed in Table 3. From Table 3, it is noticed that the best results are obtained at fc=180 at which we get 100% TPR for MICC-F220.

Table 3: Metric parameter values from applying Butterworth low pass filter and applying SIFT algorithm & forgery detection with different values of cutoff frequency

	MICC-F220				**MICC-F600**			
	TPR %	**FPR %**	**TNR %**	**FNR %**	**TPR %**	**FPR %**	**TNR %**	**FNR %**
***fc*=160**	99.09	13.64	86.36	0.91	79.38	9.35	90.64	20.62
***fc*=180**	100	5.05	94.95	0	85.5	2.7	97.3	14.5
***fc*=200**	99.09	9.09	90.91	0.91	88.75	12.68	87.32	11.25
***fc*=220**	95.45	4.54	95.46	4.55	86.25	16.18	83.82	13.75
	MICC-F2000				**SATS-130**			
	TPR %	**FPR %**	**TNR %**	**FNR %**	**TPR %**	**FPR %**	**TNR %**	**FNR %**
***fc*=160**	94.9	12.11	87.89	5.1	76.2	21.31	78.69	23.8
***fc*=180**	96.71	8.76	91.24	3.29	81.25	20.83	79.17	18.75
***fc*=200**	94.95	11.15	88.85	8.05	79.17	16.67	83.33	20.83
***fc*=220**	91.3	17.87	82.13	8.7	79.17	21.75	78.25	20.83

The previous results show that the high-pass filter has an effect in increasing both image sharpening and details of edges. The HPF allows image details to be clear and participates on declaring the difference between extracted features, but the image still has some noise and needs more smoothing which is presented by the BLPF. There is a need for a combination method mixed between image sharpening and increasing edge details from a side with noise reduction and image smoothing from the other side. A combination between HPF and BLPF is built and its results are shown in Table 4.

We combined HPF and BLPF with different cutoff frequencies from fc=160 to fc=220. The SIFT algorithm is performed with fixed values of threshold T_h = 2 and minimum number of matched features at 7 features.

The results show that the largest values of TPR and smallest values of FPR appear at fc=180 compared with other cutoff frequencies in this test. Results from HPF and BLPF combination are the best results comparing with other test methods shown in Table 1, Table 2 and Table 3 among all datasets.

Table 4: Metric parameter values after applying high pass filter first then applying Butterworth low pass filter with different values of cutoff frequencies and complete SIFT algorithm & forgery detection

	MICC-F220				**MICC-F600**			
	TPR %	**FPR %**	**TNR %**	**FNR %**	**TPR %**	**FPR %**	**TNR %**	**FNR %**
***fc*=160**	94.30	10.73	89.27	5.7	81.13	12.15	87.85	18.87
***fc*=180**	99.90	4.54	95.46	0.02	87.76	5.63	94.37	12.24
***fc*=200**	97.27	6.36	93.64	2.73	91.48	9.37	90.63	8.52
***fc*=220**	99.09	8.18	91.82	0.91	89.64	10.2	89.8	10.36
	MICC-F2000				**SATS-130**			
	TPR %	**FPR %**	**TNR %**	**FNR %**	**TPR %**	**FPR %**	**TNR %**	**FNR %**
***fc*=160**	95.2	11.83	88.17	4.8	77.53	20.15	79.85	22.47
***fc*=180**	97.18	7.65	92.35	2.82	83.18	16.72	83.28	16.82
***fc*=200**	95.29	10.89	89.11	4.71	80.27	15.86	84.14	19.73
***fc*=220**	94.13	14.76	85.24	5.87	80.39	20.57	79.43	19.61

4.4 Comparison between both the proposal and traditional methods results

In this subsection, a comparison between the proposed method results and other results in [22], [23] and [24] is presented on the same datasets in terms of the same metric parameters (TPR, FPR, TNR and FNR). Table 5 compares the values from Table 4 obtained from combination between HPF and BLPF at fc=180 with other values resulting from traditional methods. Amerini et al. [22] used MICC-F220, MICC-F2000 and MICC-F600 datasets to test their method and used MICC-F2000 and MICC-F600 datasets to test the method in paper [23]. Christlein et al. [24] used SATS-130 to test this method compared with other methods. The proposal uses the four datasets to test and compare the results as shown in Table 5.

The proposal outperforms other methods by giving higher values of TPR and mostly lower values of FPR. By comparing results in Table 5, it is clear that the proposal is a successful way to improve efficiency of copy move forgery detection based on SIFT feature extraction methods.

Table 5: Comparison between both the proposal and traditional methods results

	MICC-F220				**MICC-F600**			
	TPR %	**FPR %**	**TNR %**	**FNR %**	**TPR %**	**FPR %**	**TNR %**	**FNR %**
The proposal	100	4.54	95.46	0	91.49	9.37	90.63	8.52
Amerini et al. [22]	100	8	92	0	69.2	12.5	87.5	30.8
Amerini et al. [23]	N/A	N/A	N/A	N/A	81.6	7.27	92.73	18.4
Christlein et al. [24]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	MICC-F2000				**SATS-130**			
	TPR %	**FPR %**	**TNR %**	**FNR %**	**TPR %**	**FPR %**	**TNR %**	**FNR %**
The proposal	97.18	7.65	92.35	2.82	83.18	16.72	83.28	16.82
Amerini et al. [22]	93.42	11.61	88.39	6.58	N/A	N/A	N/A	N/A
Amerini et al. [23]	94.86	9.15	90.85	5.14	N/A	N/A	N/A	N/A
Christlein et al. [24]	N/A	N/A	N/A	N/A	79.17	11.63	88.37	20.83

4.5 JPEG Compression and Noise Adding tests

In this test, an evaluation of the proposal results are offered, but with new conditions. JPEG compression with different values of quality factors and noise adding with different values of Signal-to-Noise Ratio (SNR) over MICC-F2000 dataset are performing. Using the same values in [22], where JPEG compression quality factor varies between 100, 75, 50, 40 and 20 and SNR varies between 50, 40, 30 and 20 (dB). This test shows how this proposal withstands these preprocessing attacks and shows the quality of detecting copy move forgery under these new conditions compared with [22]. Tables 6 and 7 show the comparison between results from the proposal and method in [22]. From Table 6 and Table, 7 it is clear that the proposal gives better results compared with the other algorithms used by Amerini et al. [22]. It gives higher stability and reliability against different conditions of attacks than the other algorithms which its efficiency decreases with different values of JPEG compression factors and SNR.

Table 6: Comparison between the proposal and Amerini et al. [22] performance against different values of JPEG compression applied on whole images

JPEG Quality Factor	**The proposal**		**Amerini et al. [22]**	
	TPR%	FPR%	TPR%	FPR%
100	97.18	7.65	93.42	11.61
75	97.15	7.65	93.72	12.07
50	97.09	7.47	93.16	11.15
40	97.83	7.30	92.14	11.13
20	97.31	6.89	87.15	10.46

Table 7: Comparison between the proposal and Amerini et al. [22] performance against values of SNR (db) applied on whole images.

SNR (db)	**The proposal**		**Amerini et al. [22]**	
	TPR%	FPR%	TPR%	FPR%
50	97.18	7.65	93.71	11.46
40	97.15	7.65	94.14	11.69
30	95.37	7.21	92.00	11.46
20	93.13	6.78	82.42	8.15

4.6 Combined Attacks Tests

In this section, new datasets are fabricated by selecting 12 original images from the previous used datasets with different resolutions. We copied a patched area from each image, and then applied one type from the following combined attacks on each patched area:

a) Gaussian noise adding with SNR = 50, and then Gamma correction with value 0.7.
b) Gaussian noise adding with SNR = 50, and then JPEG compression with quality 50.
c) Gamma correction with value 0.7, and then JPEG compression with quality 50.
d) Gaussian noise with SNR = 50, then Gamma correction with value 0.7, and then JPEG compression with quality 50.

After applying the combined attacks on the patched areas, geometric transformations like that in [22] are performed on the tampered patched areas before pasting in the images. The constructed datasets contain tampered images having patched areas affected by two types of attacks. The first attack is one of the previous combined attacks. The second attack is one of the geometric transformations shown in Table 8.

where ϴ is the rotation angle, and s_x, s_y are the scaling values. Each combined attack from the previous list constructs a new dataset consisting of the twelve original images plus the twelve tampered images. The proposed algorithm is applied on each dataset also; the algorithm in [22] is applied to the same datasets and the results are shown in Table 9.

Table 8: Geometric transformations that can be applied sequentially on the tampered patched areas before pasting to the original images

Attack No.	**ϴ**	s_x	s_y	**Attack No.**	**ϴ**	s_x	s_y
1	0	1	1	7	5	1	1
2	0	0.5	0.5	8	20	1	1
3	0	0.7	0.7	9	30	1	1
4	0	1.2	1.2	10	50	1	1
5	0	1.6	1.6	11	70	1	1
6	0	2	2	12	90	1.5	1.5

Table 9: Comparison between the proposal and Amerini et al. [22] performance against different types of combined attacks applied on patched areas only

Combined Attack Type	**Algorithm**	**TPR%**	**FPR%**
Gaussian Noise with Gamma correction attack	Amerini et al. [22]	85.71	14.29
	The Proposal	100	7.14
Gaussian Noise with JPEG compression attack	Amerini et al. [22]	86.75	14.80
	The Proposal	95.06	18.30
Gamma correction with JPEG compression attack	Amerini et al. [22]	87.5	12.4
	The Proposal	93.5	14.1
Gaussian Noise with Gamma correction with JPEG compression attack	Amerini et al. [22]	87.5	12.4
	The Proposal	91.3	14.1

Table 9 shows the experimental results of applying the both algorithms on the same datasets constructed from the combination of attacks in each case with geometrical transformations with different rotation angles and different scaling values. The results show that this proposal gives higher efficiency in detecting the correctly identified forged images (TPR). But it shows that the proposal gives slightly worse results with images incorrectly identified as forged (FPR). In general, this algorithm gives good results in dealing with different types of attacks whether these attacks are individual or complex.

5. Conclusion

This paper presented enhanced copy move forgery detection algorithms based on SIFT features. Rotation, scaling, and translation are the most common geometric transformations used to mislead the forgery detection algorithms in addition to other preprocessing operations used for the same purpose such as JPEG compression, noise adding, Gamma correction, and light adjustments. The proposed approach was able to give efficient results against such forgery attacks by displaying the difference between image details. The proposed approach is recommended to work with SIFT features to give efficient forgery detection results.

References

[1] D. David, Divya B., "Image Authentication Techniques and Advances Survey", COMPUSOFT, An international journal of advanced computer technology, vol. IV, Issue IV, April 2015.

[2] D. Usha Nandini, S. Divya, "A literature survey on various watermarking techniques", Inventive Systems and Control (ICISC), 2017 International Conference on, Coimbatore, India, 19-20 January 2017

[3] Osamah M. Al-Qershi, Bee Ee Khoo, "Passive detection of copy-move forgery in digital images: State-of-the-art", Forensic Science International, 284 – 295, 3 July 2013.

[4] Judith A. Redi, Wiem Taktak, Jean-Luc Dugelay, "Digital image forensics: a booklet for beginners", Multimedia Tools Appl., vol. 51, pp. 133-162, 2011.

[5] Gajanan K. Birajdar, Vijay H. Mankar, "Digital image forgery detection using passive techniques: A survey", Digital Investigation, pp. 226-245, 2013.

[6] M. Ali Qureshi, M.Deriche, "A Review on Copy Move Image Forgery Detection Techniques", Multi-Conference on Systems, Signals & Devices (SSD), pp. 11-14, February 2014.

[7] Hany Farid, "Image Forgery Detection A survey", IEEE SIGNAL PROCESSING MAGAZINE, March 2009.

[8] M. Zimba, S. Xingming, "Fast and Robust Image Cloning Detection using Block Characteristics of DWT Coefficients", International Journal of Digital Content Technology and its Applications, vol. 5, Number 7, 2011.

[9] Manjima Mishra, Preeti Rai, "A Proposed Work on Image Forgery Detection Technique", International Journal of Computer Applications, vol. 163(2), April 2017.

[10] J. Zhao, J. Guo, "Passive forensics for copy-move image forgery using a method based on DCT and SVD", Forensic Science International, vol. 33, pp. 158-166, 2013.

[11] Seung-Jin Ryu, Matthias Kirchner, Min-Jeong Lee, and Heung-Kyu Lee, "Rotation Invariant Localization of Duplicated Image Regions Based on Zernike Moments", IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 8(8), August 2013.

[12] Muhammad Hussain, Sahar Q. Saleh, Hatim Aboalsamh, Ghulam Muhammad, George Bebis, "Comparison between WLD and LBP Descriptors for Non-intrusive Image Forgery Detection", IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA) Proceedings, pp. 197-204, Alberobello, 23 – 25 June 2014.

[13] Devanshi Chauhan, Dipali Kasat, Sanjeev Jain, Vilas Thakare, "Survey On Keypoint Based Copy-move Forgery Detection Methods On Image", Procedia Computer Science, International Conference on Computational Modeling and Security, pp. 206-212, 2016.

[14] O. M. Al-Qershi , B. E. Khoo, "Passive detection of copy-move forgery in digital images: State-of-the-art", Forensic Science International, pp. 284 – 295, 2013.

[15] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features", Proc. of the International Conference on Computer Vision, vol. 2, pp. 1150-1157, 1999.

[16] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, vol. 60(2), pp. 91-110, 2004.

[17] H. Huang, W. Guo, Y. Zhang, "Detection of Copy-Move Forgery in Digital Images Using SIFT Algorithm," IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, DOI 10.1109/PACIIA.2008.

[18] X. Pan, S. Lyu, "Region Duplication Detection Using Image Feature Matching", IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, vol. 5(4), 2010.

[19] J. Beis and D. Lowe, "Shape indexing using approximate nearest neighbor search in high dimensional spaces", Proc. of CVPR, San Juan, 1997.

[20] M. F. Hashmi, A. R. Hambarde, A. G. Keskar, "Copy Move Forgery Detection using DWT and SIFT Features", 3th International Conference on Intelligent Systems Design and Applications (ISDA), pp. 188-193, 2013.

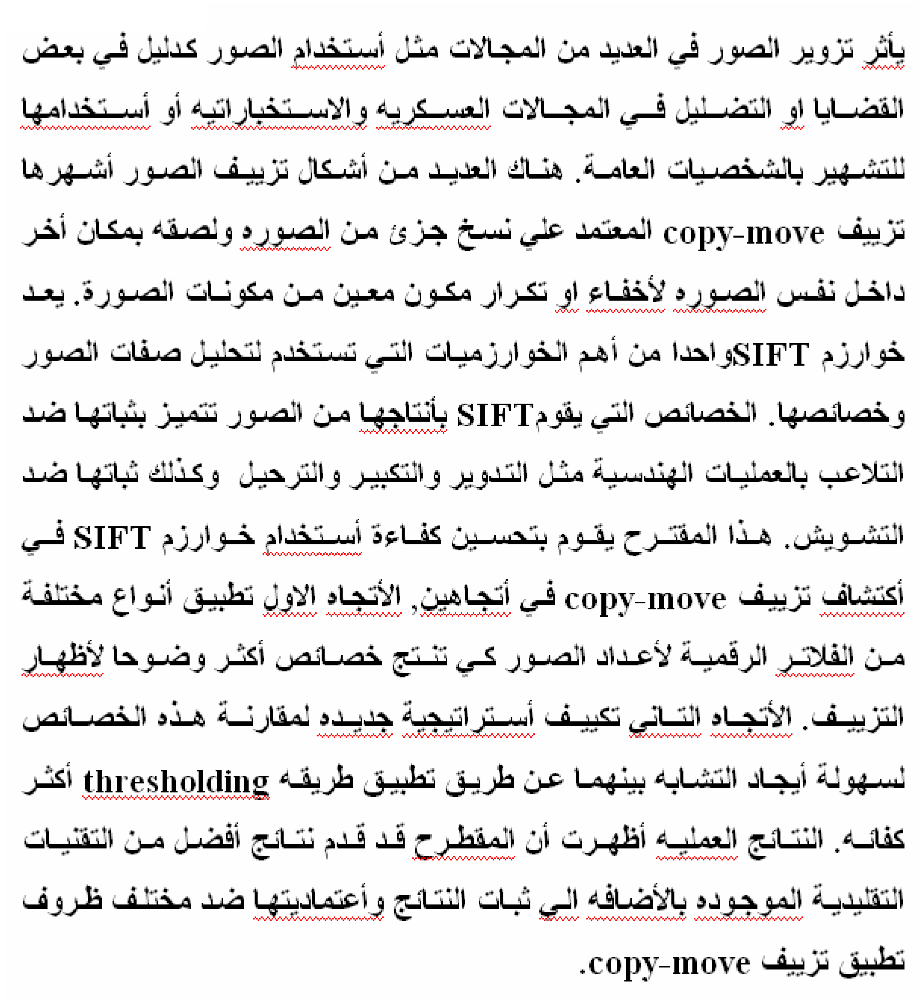
[21] T. Chihaoui, S. Bourouis, K. Hamrouni, "COPY-MOVE IMAGE FORGERY DETECTION BASED ON SIFT DESCRIPTORS AND SVD-MATCHING", 1st International Conference on Advanced Technologies for Signal and Image Processing (ATSIP'2014), Sousse, Tunisia, March 2014.

[22] I. Amerini, L. Ballan, R. Caldelli, A. D. 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(3), pp. 1099-1110, 2011.

[23] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, L. Del Tongo, and G. Serra, "Copy-Move Forgery Detection and Localization by Means of Robust Clustering with J-Linkage", Signal Processing: Image Communication, vol. 28(6), pp. 659-669, 2013.

[24] V. Christlein, C. Riess, J. Jordan, C. Riess, E. Angelopoulou: "An Evaluation of Popular Copy-Move Forgery Detection Approaches", IEEE Transactions on Information Forensics and Security, vol. 7(6), pp. 1841-1854, 2012.

الملخص باللغة العربية



يأثر تزوير الصور في العديد من المجالات مثل أستخدام الصور كدليل في بعض القضايا او التضليل في المجالات العسكريه والاستخباراتيه أو أستخدامها للتشهير بالشخصيات العامة. هناك العديد من أشكال تزييف الصور أشهرها تزييف copy-move المعتمد علي نسخ جزئ من الصوره ولصقه بمكان أخر داخل نفس الصوره لأخفاء او تكرار مكون معين من مكونات الصورة. يعد خوارزم SIFTواحدا من أهم الخوارزميات التي تستخدم لتحليل صفات الصور وخصائصها. الخصائص التي يقومSIFT بأنتاجها من الصور تتميز بثباتها ضد التلاعب بالعمليات الهندسية مثل التدوير والتكبير والترحيل وكذلك ثباتها ضد التشويش. هذا المقترح يقوم بتحسين كفاءة أستخدام خوارزم SIFT في أكتشاف تزييف copy-move في أتجاهين, الأتجاه الاول تطبيق أنواع مختلفة من الفلاتر الرقمية لأعداد الصور كي تنتج خصائص أكثر وضوحا لأظهار التزييف. الأتجاه التاني تكييف أستراتيجية جديده لمقارنة هذه الخصائص لسهولة أيجاد التشابه بينهما عن طريق تطبيق طريقه thresholding أكثر كفائه. النتائج العمليه أظهرت أن المقطرح قد قدم نتائج أفضل من التقنيات التقليدية الموجوده بالأضافه الي ثبات النتائج وأعتماديتها ضد مختلف ظروف تطبيق تزييف copy-move.