International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 30, No. 3 (2016) 1655008 (23 pages)

DOI: 10.1142/S0218001416550089

Keypoints-Based Image Passive Forensics Method for Copy-Move Attacks

Xiaofeng Wang*, Guanghui He, Chao Tang, Yali Han
and Shangping Wang

School of Science, Xi'an University of Technology
Xi'an, Shaanxi 710048, P. R. China
**xfwang66@sina.com.cn*

Received 26 July 2014
Accepted 20 October 2015
Published 12 January 2016

A novel image passive forensics method for copy-move forgery detection is proposed. The proposed method combines block matching technology and feature point matching technology, and breaks away from the general framework of the visual feature-based approach that used local visual feature such as SIFT and followed by a clustering procedure to group feature points that are spatially close. In our work, image keypoints are extracted using Harris detector, and the statistical features of keypoint neighborhoods are used to generate forensics features. Then we proposed a new forensics features matching approach, in which, a region growth technology and a mismatch checking approach are developed to reduce mismatched keypoints and improve detected accuracy. We also develop a duplicate region detection method based on the distance frequency of corresponding keypoint pairs. The proposed method can detect duplicate regions for high resolution images. It has higher detection accuracy and computation efficiency. Experimental results show that the proposed method is robust for content-preserving manipulations such as JPEG compression, gamma adjustment, filtering, luminance enhancement, blurring, etc.

Keywords: Passive forensics; copy-move attacks; duplicate regions detection; region growth; feature matching.

1. Introduction

With the development of multimedia processing technology and the using of the powerful image processing software, it is easy to tamper digital images without leaving any traces. Many image tampering issues indicate that "seeing is believing" is not true. Figure 1 shows the famous image tampering events occurred in recent years. Sophisticated image forgery events have caused the public suspicion of multimedia data and have raised a number of important security challenges to the forefront.

As one of the important technologies of protecting digital image content authenticity, image passive forensics[1,2] has been developed in recent years. It is a

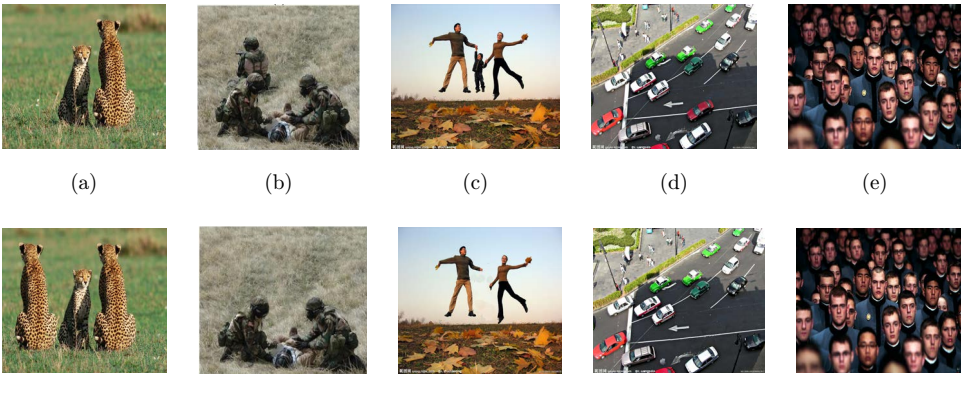
(a) (b) (c) (d)

(a_1) (b_1) (c_1) (d_1)

Fig. 1. (a)–(d) are forged images; (a_1), (b_1) and (d_1) are detected results and (c_1) is an original image.

technology that detecting whether an image has subjected to certain manipulation or not without relying on any prior information about the original image. It is accomplished by analyzing the intrinsic traces left by devices and processing and identifying inconsistencies signal characteristics.[3–5] Due to do not need any prior information, image passive forensics technology has extensive application prospects in practice.

The image forensics method for copy-move forgery detection has attracted great interest in recent years because the copy-move forgery is the most common image tampering approach. In a copy-move forgery, a part of image content is copied and pasted somewhere else in the same image with the intent to conceal image objects or clone some regions to produce a non-existing scene (see Fig. 2). Therefore, the



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

(a_1) (b_1) (c_1) (d_1) (e_1)

Fig. 2. Some examples of copy-move forgery. (a)–(e) are original images and (a_1)–(e_1) are tampered images.

existence of similar regions in image can be considered as the evidence to trace tampering or forging. However, it is very difficult to find the very same regions in image because the forgery will likely be saved by post-processing, in which, the possible using of the retouch tool or other localized image processing tools will lead to the mismatch between cloned regions and the original regions.[6]

1.1. *Previous works*

Taking a general survey on image passive forensics technology, existing methods for copy-move forgery detection can be classified into two main categories: blocks matching methods and feature point matching methods.

Many of the early duplication regions detection methods are based on image blocks matching. The general processes of these methods are as follows: tested image is partitioned into overlapping or non-overlapping blocks and the features of each block are extracted and sorted, and then, cloned blocks could be detected by matching the features of each image blocks. Reference 6 suggested first such method. In this that work, a 8×8 image block was slid pixel-by-pixel from the left top to the right bottom of an image, and then the quantified DCT coefficients of each block were used as features. The features of all blocks were sorted lexicographically, and then they checked whether the adjacent blocks were similar or not by matching their features. This method can detect cloned regions in image, however, since high computational complexity, many subsequent works such as Refs. 7–12 focused on improving calculation efficiency. Reference 7 employed the principle component analysis (PCA) to reduce the feature dimension. Reference 8 combined discrete wavelet transformation with singular value decomposition (SVD) to generate feature vectors, and then the feature vectors were matched to detect clone regions. Recent research, Ref. 13, presented an efficient expanding block algorithm for image copy-move forgery detection. This method primarily uses direct block comparison rather than indirect comparisons based on block features. The advantage of direct block comparison is it can be done without a large sacrifice in performance time. Reference 14 compared four block-based detection methods for copy-cover forgery detection, which are based on PCA, DCT, spatial domain, and statistical domain. It is concluded that the PCA method outperforms the others in terms of time complexity and accuracy.

Blocks matching techniques really provide an efficient approach to detect cloned image regions, but almost of them have some common shortages such as high computational complexity and weaker robustness for content-preserving image manipulations such as JPEG compression, gamma adjustment, filtering, luminance enhancement, blurring, and so on. Therefore, some alternative methods have been explored in recent years, in which, detection approaches based on feature point matching have been paid close attention. In this kind of methods, scale-invariant feature transform (SIFT)[15] was extensively used to generate forensics feature, also, SIFT matching was often followed by a clustering procedure to group keypoints that

are spatially close. Reference 16 suggested first such a method to detect copy-move forgery. In Ref. 16, the SIFT keypoints were matched each other to seek for any possible duplication regions in images. Reference 17 proposed a method that used SIFT keypoints matching to estimate the parameters of the affine transformation. Similar to Ref. 16, it cannot exactly detect the locations of duplicated regions, but only give a vague description about the outline of tampered regions by displaying very few matched keypoints. Subsequently, the same authors proposed a detection method (see Ref. 18) that understood if a copy-move attack has occurred and, furthermore, to recover the geometric transformation used to perform cloning. Very recently, they presented an efficient method (Ref. 19) that improved their previous works (Refs. 17 and 18) by introducing a new robust clustering phase based on the J-Linkage algorithm,[20] which performed a robust clustering in the space of the geometric transformation. Reference 21 presented a SIFT-matching-based method that can detect duplicated regions with rotation or scaling, in which, 128-dimensional SIFT feature vector was used to find all pixels within the duplicated regions after discounting the estimated transforms. Reference 22 also reported a technique that extracted local feature by using SIFT, and matched features by applying best-bin-first method,[23] then performed clustering to find the duplicate regions. Their method is efficacy to detect copy-paste forgeries with translation, scaling, rotation, flipping, lossy compression, noise addition, and blurring. Reference 24 reported a method that extracted local feature by using SIFT, and matched local features by using KD-tree and best-bin-first method. Reference 25 proposed a duplication region detection method, in which, Harris corners were detected first, then the step sector statistics were developed to represent the small circle image region around each Harris point with a feature vector. Finally, the small circle image regions were matched using the best-bin-first algorithm to reveal duplicate regions.

In order to improve the efficiency of detection algorithm, Refs. 26 and 27 used speed up robust features (SURF) to construct features for copy-move forgery detection. To improve the robust of the algorithm, some image statistical features were used to detect copy-move attacks, e.g. Ref. 16 used chroma and illuminance information, Ref. 12 used the statistical value of the wavelet coefficients, Ref. 10 extracted the invariant blur moment of image, and Refs. 28 and 29 used log-polar transformation coefficients, and so on.

1.2. *Existing problems*

Taking a general survey on image copy-move forgery detection, there are some negative problems existed in existing methods, we drawing out main problems as follows:

(1) Lower detection accuracy (see Figs. 3–5). Figure 3 shows an example of a detection method based on block matching. We can see that the detected regions in (c) are larger than actually object. Figures 4 and 5 show the examples of two

(a) (b) (c)

Fig. 3. An example of detected copy-move regions in (c) are larger than object region (Ref. 9). (a) Original image, (b) tampered image, (c) detected result.

(a) (b) (c)

Fig. 4. A few of keypoints are detected, the cloned regions could not be found via these points (Ref. 30). (a) Original image, (b) tampered image, (c) detected result.

feature point-based methods, as can be seen, only a few keypoints are detected and matched, and they could not cover the cloned regions. Moreover, Fig. 5 shows error detection that will result a high false positive rate.

(2) SIFT-based methods can efficiently detect duplicated and distorted image regions via matching feature points. However, they cannot find reliable keypoints in some regions of smooth texture that include little visual structures.

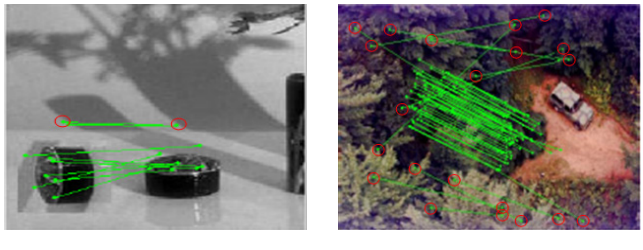


Fig. 5. A few of keypoints cannot cover cloned regions; the keypoints in red circles are error detection.[16]

Fig. 6. Top left image is a forgery, in which, a duplicated region cannot be detected via using Ref. 21 due to the lack of reliable keypoints. Other three images are not tampered, but the intrinsic repetitive patterns are regarded as duplicated regions via using Ref. 21.

Therefore, many such methods are often unavailable for the top left image in Fig. 6, where an obvious duplicated region cannot be detected.[21] Similarly, for smaller duplicated regions, as fewer keypoints, they are also hard to be detected using such a method. Additionally, some images that include intrinsically identical regions cannot be differentiated from intentionally inserted duplicated regions,[21] Fig. 6 exemplify such cases.

1.3. *Our contribution*

In this paper, a new method is developed to detect copy-move forgery. Our work solves above mentioned issues via combining blocks matching technology and feature points matching technology. In the proposed method, Harris detector is used to extract keypoints, and a square region for each keypoint is generated. The statistical values of DCT coefficients from these regions are used to generate forensic features. Then a new feature matching method is developed, in which, the distance between each pair of keypoints is measured, and the frequencies of these distances are used to estimate the duplicate regions. Experimental results show that the proposed method is provided with satisfactory detection accuracy, detection efficiency, and robustness. It can be used to detect duplicated regions for high resolution images.

The remainder of this paper is organized as follows: The proposed method is described in Sec. 2. Experimental results are shown in Sec. 3. Finally, a brief conclusion is summarized in Sec. 4.

2. Proposed Method

Our goal is to understand if a tested image contains duplicated regions. In a copy-move forgery, a part of image content is segmented, copied and pasted into somewhere of the same image, which will introduce a correlation between the original region and the duplicated one. This correlation can be used to detect the forgery. Considering the most general case, a copy-move attack can be regard as an image region is shifted certain distance in the same image to generate a forged region, thus the original region and cloned region include same pixel points. Therefore, if there is a copy-move forgery in an image, then there must be two corresponding pixel points' sets, in which, every pair of corresponding pixel points have identity distance. Motivated by this idea, we developed a feature extraction and duplicated regions detection algorithm. Our method includes three stages: forensics feature extraction, forensics features matching and duplicate regions detection. The framework is shown in Fig. 7.

2.1. *Forensics feature extraction*

We start our work from finding similar keypoints in image. Let $f(x, y)$ denote a $M \times N$ image, we use the Harris detector[31] to extract keypoints from $f(x, y)$. Let A denote keypoints set:

$$A = \{A_1(x_1, y_1), A_2(x_2, y_2), \ldots, A_n(x_n, y_n)\}, \tag{1}$$

where (x_i, y_i) is the position coordinate of A_i, n is the number of keypoints.

2.1.1. *Keypoint neighborhood*

Considering each keypoint is associated with its surroundings, we define a neighborhood for each keypoint. For the keypoint $A_i(x_i, y_i) \in A$ $(1 \leq i \leq n)$, define a square neighborhood NA_i with center at $A_i(x_i, y_i)$ and the size is $d \times d$, we call it keypoint neighborhood. Here, the selection of d is according to the statistics value

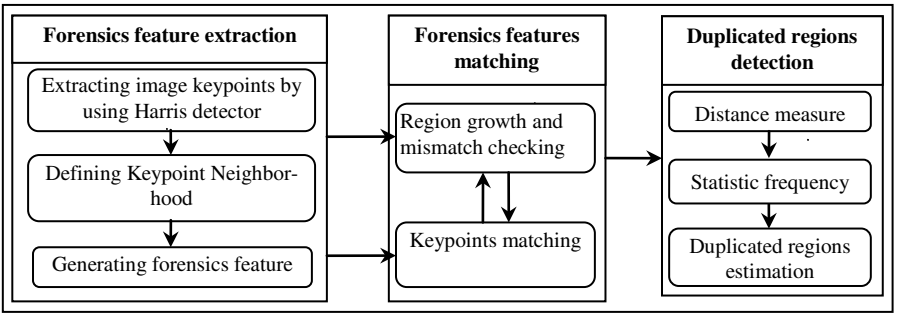


Fig. 7. The framework of the proposed method.

(see Sec. 3.5). The kth keypoint neighborhood is denoted as:

$$NA_k(x_k, y_k) = \begin{pmatrix} a_{11}^k & a_{12}^k & \dots & a_{1d}^k \\ a_{21}^k & a_{22}^k & \dots & a_{2d}^k \\ \dots & \dots & \dots & \dots \\ a_{d1}^k & a_{d2}^k & \dots & a_{dd}^k \end{pmatrix}, \tag{2}$$

where x_k, y_k are position coordinates of keypoint, a_{st}^k $(s,\ t = 1,2,\dots,d)$ are pixel values.

2.1.2. *Forensics feature*

In order to generate forensics features, we compute the statistical values of the keypoint neighborhood. Applying DCT for each keypoint neighborhood $NA_k(x_k, y_k)$, and the coefficient matrix is denoted as:

$$M_k = \begin{pmatrix} b_{11}^k & b_{12}^k & \dots & b_{1d}^k \\ b_{21}^k & b_{22}^k & \dots & b_{2d}^k \\ \dots & \dots & \dots & \dots \\ b_{d1}^k & b_{d2}^k & \dots & b_{dd}^k \end{pmatrix}, \tag{3}$$

where

$$b_{st}^k = \begin{cases} \mathrm{DC}_k, & \text{if } (s = 1) \wedge (t = 1) \\ \mathrm{AC}_{st}^k, & \text{if } (s \in \{1,\dots,d\}) \wedge (t \in \{1,\dots,d\}) \wedge ((t \neq 1) \vee (s \neq 1)) \end{cases} \tag{4}$$

where DC represents the direct current component of the DCT coefficients and AC represents the alternating current components of the DCT coefficients.

Calculate the mean of matrix M_k, and denoted as

$$m_k = \frac{1}{d \times d}\left(\sum_{s=1}^{d}\sum_{t=1}^{d} b_{st}^k\right). \tag{5}$$

Let DC_k and m_k as forensics features, store DC_k, m_k, and corresponding keypoints coordinates x_k and y_k into a matrix MF, where $k = 1,2,\dots,n$.

$$\mathrm{MF} = \begin{pmatrix} \mathrm{DC}_1 & m_1 & x_1 & y_1 \\ \dots & \dots & \dots & \dots \\ \mathrm{DC}_k & m_k & x_k & y_k \\ \dots & \dots & \dots & \dots \\ \mathrm{DC}_n & m_n & x_n & y_n \end{pmatrix} \tag{6}$$

2.2. *Forensics features matching*

In copy-move forgery, a part of image content is copied and pasted somewhere else in the same image. Therefore, if there is a copy-move forgery in an image, then there must be more than one region that contains similar keypoints. In this section, we first seek for similar keypoints via using the matrix MF, and then detect the duplicated

regions via using the correlation of the similar keypoints. We develop our duplicated regions detection algorithm as follows.

2.2.1. *Keypoint matching*

Randomly select MF_s and MF_t from MF, denoted as

$$\mathrm{MF}_s = (\mathrm{DC}_s, m_s, x_s, y_s), \quad \mathrm{MF}_t = (\mathrm{DC}_t, m_t, x_t, y_t). \tag{7}$$

Check the following limiters (8) and (9):

$$|\,\mathrm{DC}_s - \mathrm{DC}_t\,| < \mathrm{Threshold}_1, \tag{8}$$

$$|\,m_s - m_t\,| < \mathrm{Threshold}_2. \tag{9}$$

If both (8) and (9) are true, then MF_s and MF_t are a pair of matched keypoints, store them into a matrix M_1. Here, $\mathrm{Threshold}_1$ and $\mathrm{Threshold}_2$ are obtained by experimental statistical value (see Sec. 3.6). The matrix M_1 can be described as

$$M_1 = \begin{pmatrix} x_s & y_s & x_t & y_t \\ x_{s_1} & y_{s_1} & x_{t_1} & y_{t_1} \\ \cdots & \cdots & \cdots & \cdots \\ x_{s_l} & y_{s_l} & x_{t_l} & y_{t_l} \end{pmatrix}. \tag{10}$$

2.2.2. *Region growth and mismatch checking*

In M_1, each row represents a pair of matched keypoints. To eliminate mismatching keypoints from M_1, we develop a region growth and mismatch checking algorithm. Supposing (x_s, y_s) and (x_t, y_t) are a pair of matched keypoints, we right shift them pixel by pixel, then use new points as centers to generate two new neighborhoods $NA_s(x_s, y_s + 1)$ and $NA_t(x_t, y_t + 1)$, respectively, see Fig. 8.

$$\mathrm{NA}_s(x_s, y_s) = \begin{pmatrix} a^s_{11} & a^s_{12} & \cdots & a^s_{1d} \\ a^s_{21} & a^s_{22} & \cdots & a^s_{2d} \\ \cdots & \cdots & \cdots & \cdots \\ a^s_{d1} & a^s_{d2} & \cdots & a^s_{dd} \end{pmatrix} \xrightarrow{\text{Right shift one pixel}} \mathrm{NA}_s(x_s, y_s + 1) = \begin{pmatrix} a^s_{12} & a^s_{13} & \cdots & a^s_{1d+1} \\ a^s_{22} & a^s_{23} & \cdots & a^s_{2d+1} \\ \cdots & \cdots & \cdots & \cdots \\ a^s_{d2} & a^s_{d3} & \cdots & a^s_{dd+1} \end{pmatrix},$$

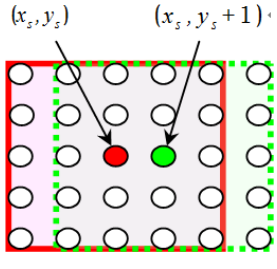


Fig. 8. The rough sketch of the neighborhood changing via right shift one pixel.

$$\mathrm{NA}_t(x_t, y_t) = \begin{pmatrix} a_{11}^t & a_{12}^t & \dots & a_{1d}^t \\ a_{21}^t & a_{22}^t & \dots & a_{2d}^t \\ \dots & \dots & \dots & \dots \\ a_{d1}^t & a_{d2}^t & \dots & a_{dd}^t \end{pmatrix} \xrightarrow{\text{Right shift one pixel}} \mathrm{NA}_t(x_t, y_t + 1) = \begin{pmatrix} a_{12}^t & a_{13}^t & \dots & a_{1d+1}^t \\ a_{22}^t & a_{23}^t & \dots & a_{2d+1}^t \\ \dots & \dots & \dots & \dots \\ a_{d2}^t & a_{d3}^t & \dots & a_{dd+1}^t \end{pmatrix}.$$

We use the method described in Sec. 2.2.1 to match new point pair. If they satisfy the limited conditions (8) and (9), then store them into matrix M_2. Otherwise, delete this pair of pixel points as well as according keypoints pair from M_1. Next, right shift to next matched keypoints, implements above process again, and stop this process until traversing M_1. Our algorithm can be described in Fig. 9.

$$M_2 = \begin{pmatrix} x^k & y^k & x^j & y^j \\ x^k & y^{k+1} & x^j & y^{j+1} \\ \dots & \dots & \dots & \dots \\ x^{k_{n'}} & y^{k_{n'}} & x^{j_{n'}} & y^{j_{n'}} \end{pmatrix}. \tag{11}$$

After finishing right shift search, the similar process is implemented via left shifting keypoints in M_1, and use the checked results to update M_2. Similarly, the similar processes are implemented via upward shifting and downward shifting the points of M_2, obtain all matched points and store the coordinates into the

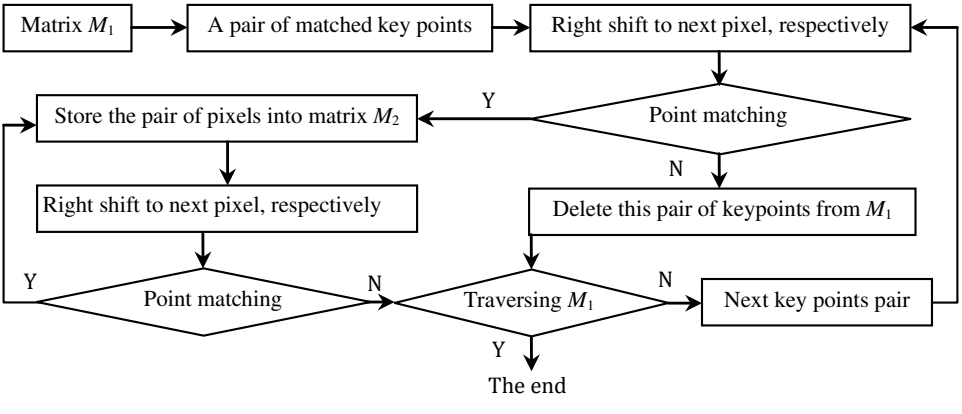


Fig. 9. The flowchart of the region growth and keypoint mismatch checking algorithm.

final matrix M_f

$$M_f = \begin{pmatrix} x_1 & y_1 & x'_1 & y'_1 \\ \cdots & \cdots & \cdots & \cdots \\ x_k & y_k & x'_k & y'_k \\ \cdots & \cdots & \cdots & \cdots \\ x_n & y_n & x'_n & y'_n \\ \cdots & \cdots & \cdots & \cdots \end{pmatrix}. \tag{12}$$

2.3. *Duplicated regions detection*

In order to detect duplicated regions, we measure the Euclidean distance of each pair of points in matrix M_f:

$$\mathrm{Ed}_k = \sqrt{(x_k - x'_k)^2 + (y_k - y'_k)^2}. \tag{13}$$

Cloned regions can be considered as moving a set of points to somewhere else in the same image, therefore, compare the original region with the duplicated region, the Euclidean distances of each pair of corresponding points are almost same except a few error matching (see Fig. 10). In order to demonstrate this fact, we count the number of distances between corresponding point pairs in M_f for Fig. 11, and calculated the frequency of the same distance values. The statistical values are shown in Table 1. Here, $N(\mathrm{Ed}_k)$ denotes the number of point pairs whose distance equal to Ed_k,

$$P(\mathrm{Ed}_k) = \frac{N(\mathrm{Ed}_k)}{\text{Total row number of matrix } M_f}. \tag{14}$$

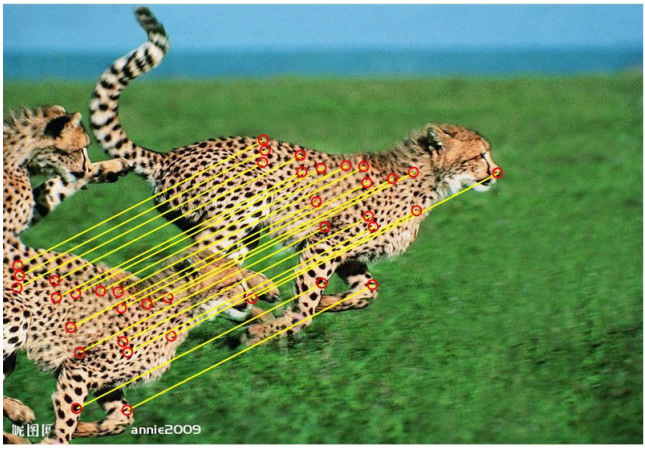


Fig. 10. Comparing original region with duplicated region, the Euclidean distances of each pair of corresponding points are almost same.

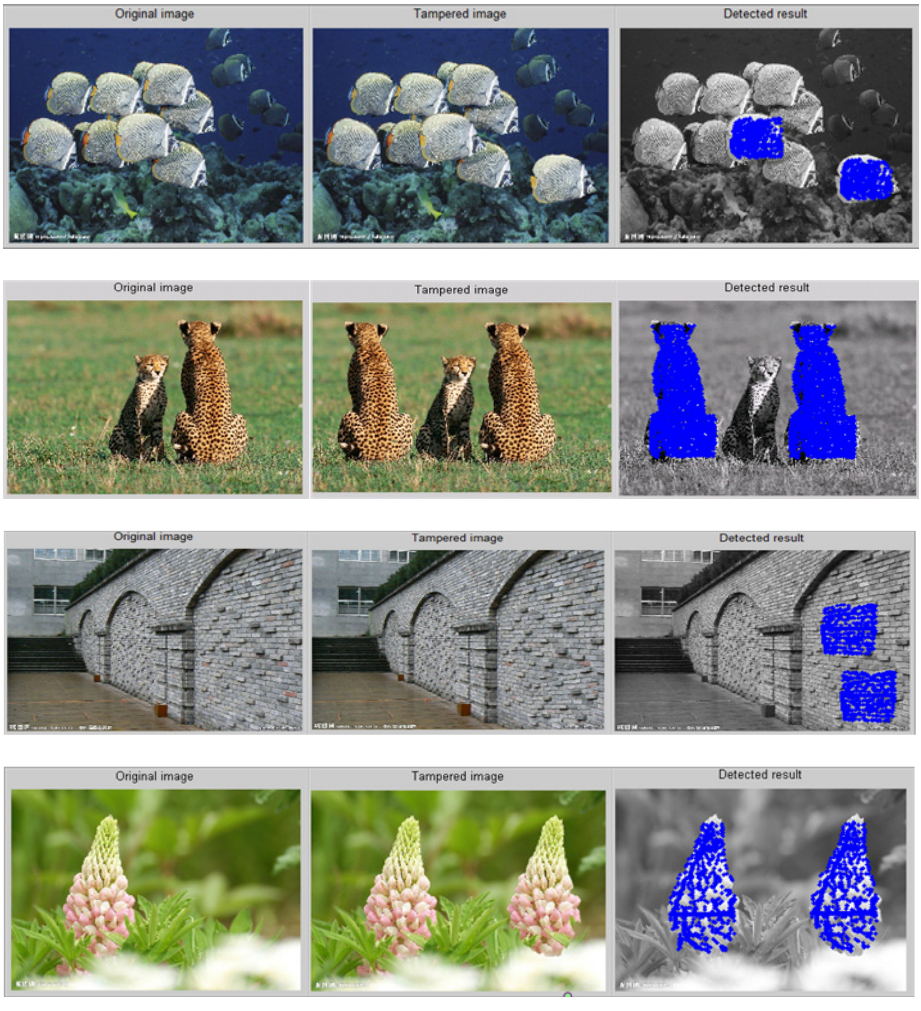


Fig. 11. The detection results for copy-move attacks.

Table 1. The frequency of distances between corresponding point pairs in M_f.

Ed_k	437.90	142.30	83.80	84.30	102.50	75.80	210.00	247.20
$P(\mathrm{Ed}_k)$ (%)	97.30	0.60	0.50	0.50	0.50	0.25	0.25	0.12

As can be seen from Table 1, there are about 97.3% of the point pairs have identical distance. We use this distance as a distance measure to estimation duplicated points set. We denote this distance as Ed_0. That is:

$$\mathrm{Ed}_0 = \arg(\max_k(P(\mathrm{Ed}_k))). \tag{15}$$

3. Experimental Results and Analysis

In this section, we discuss the performance of the proposed algorithms via experiments. Our method is implemented and tested using MATLAB2010a and famous stirmark benchmark. We run programs in the computer with Processor Pentium(R) Dual-Core CPU, i5-2400 @ 3.10 GHz, 2.00 GB RAM.

3.1. *The visual effects to detect copy-move forgeries*

These experiments are designed to test the proposed method is perceptually valid for copy-move forgery detection. To this end, we firstly tampered each tested image via copying its part content and pasting to somewhere else in the same image. To be practical, we clone image objects as duplicated regions rather than regular square image blocks, and then detect copy-move forgeries using proposed approach.

We took the size of keypoint neighborhood as 33×33, that is, $d = 33$. Here, the estimation of parameter d, thresholds Threshold_1 and Threshold_2 were according to the statistical value in Sec. 3.6. The experimental results are shown in Figs. 11–14. Figure 11 shows the detection results for copy-move attacks. Figure 12 shows the detection results for small region copy-move attacks. Figure 13 shows the detection results for multi-regions copy-move forgeries. Figure 14 shows the detected results for

Fig. 12. The detection results for small region copy-move attacks.

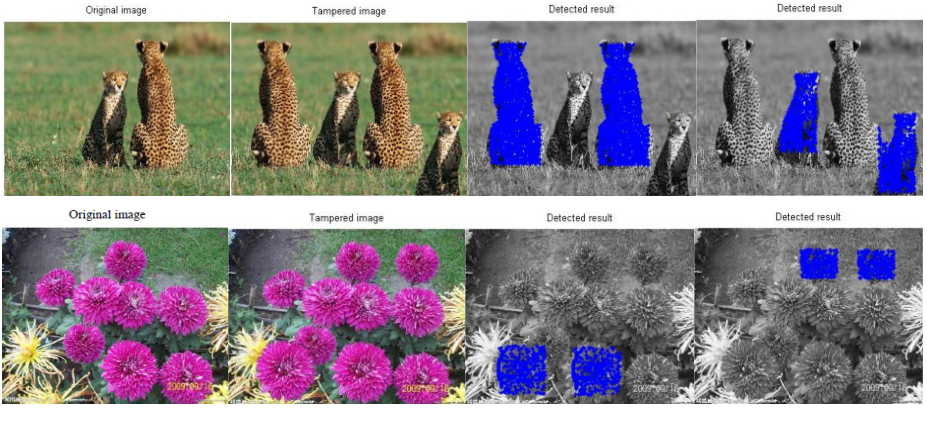


Fig. 13. The detection results for multi-regions copy-move attacks.

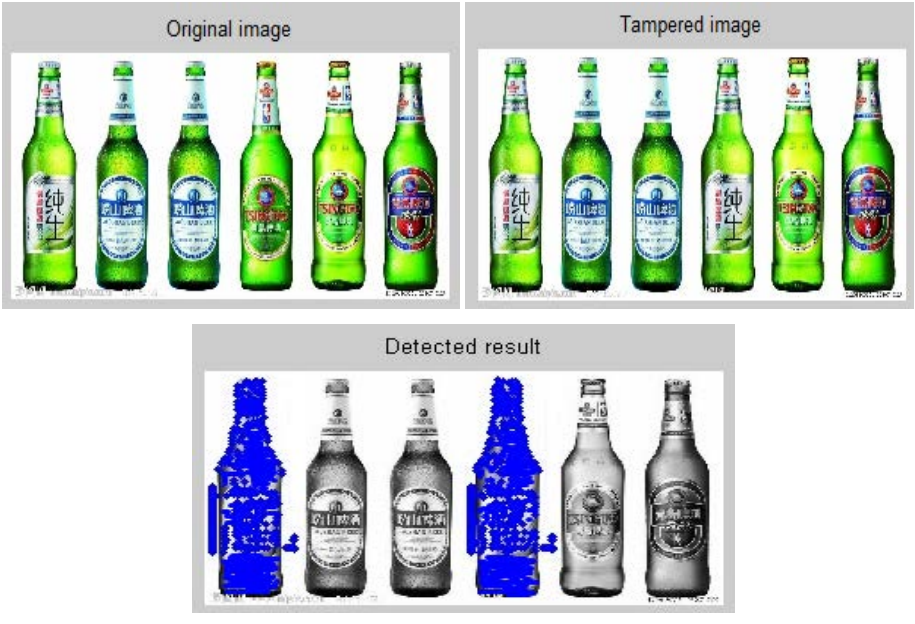


Fig. 14. In "original image", the second bottle and the third bottle are intrinsically identical regions. The "tampered image" is a tampered version of "original image" through replacing the fourth bottle by the first one. "detected result" shows that proposed method can distinguish duplicated region from intrinsically identical regions.

distinguishing duplicated region from intrinsically identical regions. Additionally, Fig. 11 (fish, wall) and Fig. 12 also include intrinsically identical regions.

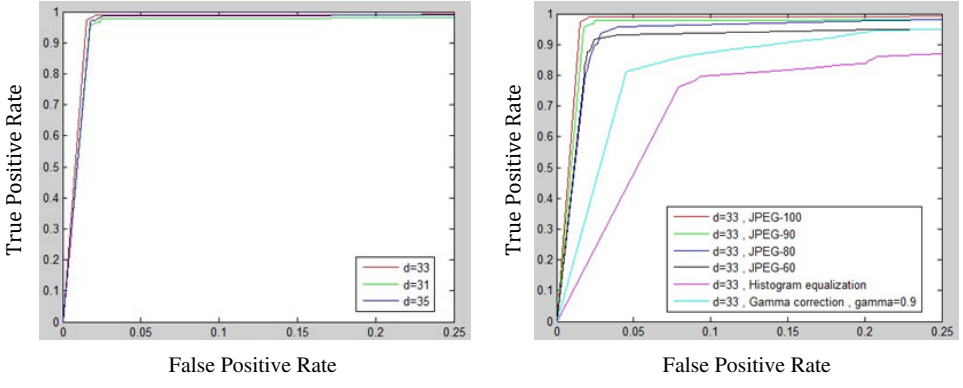
3.2. *Performance analysis with ROC*

In order to investigate the performance of the proposed method, we analyze it in terms of the receiver operating characteristics (ROCs). ROC is a plot of the false positive rate (P_{FR}) versus the true positive rate (P_{TR}) as the system parameters are varying. In context to our system, P_{FR} and P_{TR} are defined as follows:

$$P_{\mathrm{TR}} = \text{True positive rate} = \frac{\text{The number of forged images are detected as forged}}{\text{Total number of forged images}} \times 100\%, \quad (16)$$

$$P_{\mathrm{FR}} = \text{False positive rate} = \frac{\text{The number of authentic images are detected as forged}}{\text{Total number of authentic images}} \times 100\%. \quad (17)$$

To estimate P_{FR} and P_{TR}, we conduct experiments using a data set including 1000 authentic images and 1000 forged images. According to statistics, an image is considered as including similar regions if these regions include at least 20 pairs matching



(a) (b)

Fig. 15. The ROCs curve of the proposed method, where, $\text{Threshold}_1 = 11$, and Threshold_2 changing from 0.27 to 0.45. (a) Comparison of ROC for different sizes of feature point neighborhoods and (b) comparison of ROC for various image processing, including JPEG compression with quality factor 60, 80, 90, and 100, histogram equalization, and gamma correction with factor 0.9.

feature points. In experiments, we estimated P_{FR} and P_{TR} for original tested images and manipulated images via JPEG compression, histogram equalization, and gamma correction, respectively, as the system parameters $\text{Threshold}_1 = 11$, and Threshold_2 changing from 0.27 to 0.45, and then we obtained the ROC curves (see Fig. 15).

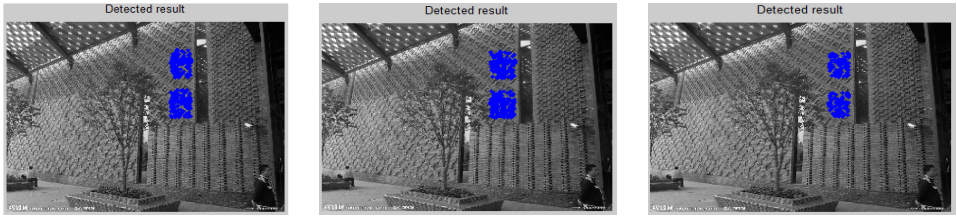
As can be seen from Fig. 15, our method is able to reach high accuracies at low false positive rates. In the case of JPEG compression, it can be seen that the performance tends to suffer with decreasing quality factor. Moreover, P_{FP} and P_{TP} are acceptable even with histogram equalization, and gamma correction.

3.3. *Robustness against incidental changes*

This experiment is designed to investigate the robustness of the proposed method under different incidental changes caused by content preserving manipulations, such as JPEG compression, histogram equalization, gamma correction, as well as luminance enhancement and blur. In the experiment, we tamper tested images via copy-move attack, and then manipulated tampered images via JPEG compression, histogram equalization, gamma correction, luminance enhancement and blur, respectively. We detected manipulated images using the proposed method, and some examples of the experimental results are shown in Figs. 16–19. As can be seen from these results, the proposed method is valid even if the tampered images have subjected to various content preserving manipulations. This means that the proposed method is robust for these content preserving manipulations.

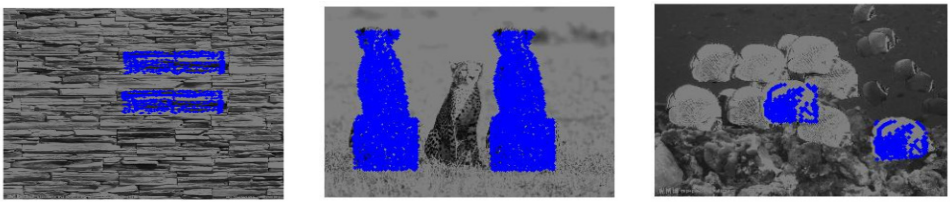
3.4. *Efficiency testing*

We evaluate the efficiency of the proposed method via testing the computing time for duplicated regions detection. In experiment, we test 6 groups images with different



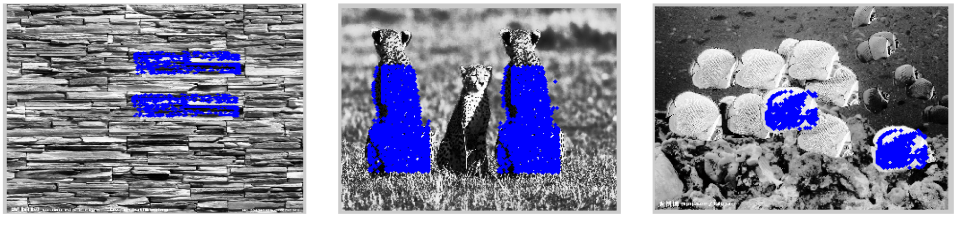
(a) (b) (c)

Fig. 16. The examples of detected results for copy-move attacked images that have subjected to JEPG compress with quality 80%, 50%, and 40%. (a) JEPG80, (b) JEPG50, (c) JEPG40.



(a) Detected result (b) Detected result (c) Detected result

Fig. 17. The examples of detected results for copy-move attacked images that have subjected to gamma correction with gamma = 0.9.



(a) Detected result (b) Detected result (c) Detected result

Fig. 18. The detected results for copy-move attacked images that have subjected to histogram equalization.

sizes. Per group includes 100 images. For each tested image, the sizes of forged regions are from 100×100 to 300×300. Table 2 shows the statistical average values of the time cost. In Table 2, the top row indicates the sizes of tested images; the most left column indicates the sizes of the forged regions; the remainders are computing time. As can be seen from Table 2, the proposed method is efficient in computing time.

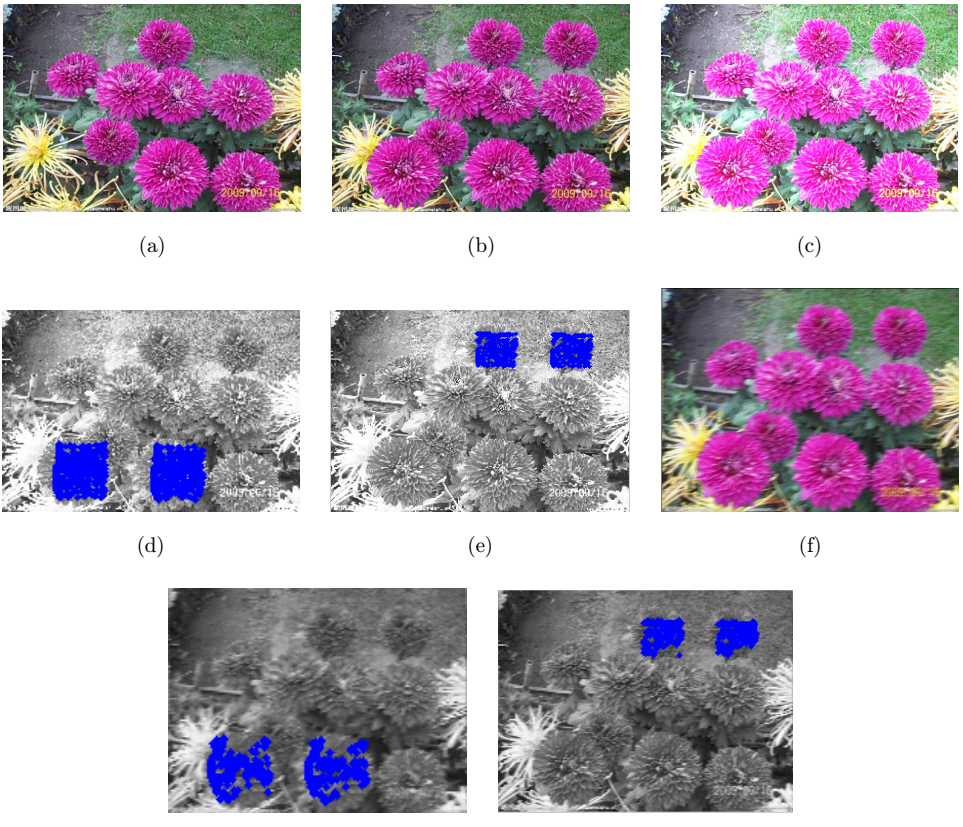


Fig. 19. The examples of detected results for copy-move attacked images that have subjected to Luminance enhancement and blur processing. (a) Original image, (b) forged image, (c) luminance enhancement of (b), (d) detected result 1 of (c), (e) detected result 2 of (c), (f) the blurring version of (b), (g) the detected result 1 of (f), (h) the detected result 2 of (f).

Table 2. Time cost (s) for detecting different sizes of forged regions.

	Image Size					
Forged Region Size	424×424	512×512	656×656	800×600	832×832	1024×768
300×300	17.2658	21.2351	27.3892	25.2519	24.9238	25.5266
280×280	16.4922	19.2660	25.1583	23.4610	24.9736	22.9128
270×270	16.1603	17.3549	23.2437	20.2253	23.1826	20.1661
250×250	13.2324	17.2603	22.1860	18.3819	23.1073	17.2616
200×200	11.2137	15.1273	19.2531	17.2921	20.1631	14.6628
190×190	10.9825	13.2730	15.2152	13.2604	20.0035	13.2704
160×160	10.2483	11.2721	10.1537	12.1796	17.1900	10.1168
130×130	6.1016	6.9143	9.1022	11.6531	16.2801	9.1510
120×120	4.2872	6.3946	8.2374	11.3461	11.9732	8.8701
100×100	3.3401	5.2102	5.3719	10.0620	11.0212	8.3525

3.5. *Comparisons*

We compared our method with several existing methods in two aspects:

(1) *Keypoints extraction*. We compared the Harris detector with other keypoints extraction methods (SIFT and SURF) in term of computing efficiency. Tested images are shown in Fig. 20. We extract same number of keypoints using Harris detector, SIFT and SURF, and test their computing time, respectively, results are shown in Table 3. As can be seen that the Harris detector is faster than SIFT and SURF.

(2) We compare the performance of the proposed method with some existing methods in term of numerical metrics. To make uniform, the size of the tested image is 1024×768. Considering the size of the keypoint neighborhood is 33×33 in our method, to compare, we take the size of block as 33×33 in Refs. 6–8 and 24. The comparison results are shown in Table 4, here, $N = 33$.

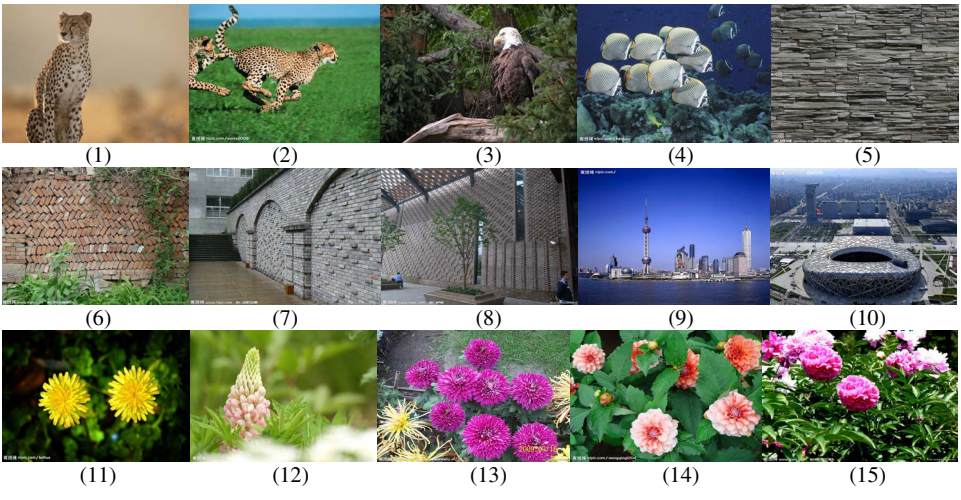


Fig. 20. Tested images.

Table 3. The computing time (s) of keypoint extraction.

Image Index	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
Computing time by Harris detector (s)	3.9	1.9	11.5	4.4	6.6	7.2	6.9	8.2	3.1	6.9	1.9	1.7	7.2	3.6	6.0
Computing time by SIFT (s)	8.7	3.8	31.2	10.2	13.5	16.2	14.7	19.9	6.0	9.6	3.3	3.3	17.8	6.6	15.0
Computing time by SURF (s)	7.2	3.3	23.8	6.9	12.7	12.4	10.9	11.3	5.8	9.4	4.2	3.8	12.3	6.9	11.7

Table 4. The comparison results.

Numerical Metrics	Ref. 6	Ref. 7	Ref. 8	Ref. 24	The Proposed Method
Feature representation	DCT	PCA	SVD	SIFT, KD-tree	DCT, keypoints
Blocked mode and the number of blocks	Overlapping blocks $(1024-33+1)\times(768-33+1)=730112$	Overlapping blocks $(1024-33+1)\times(768-33+1)=730112$	Overlapping blocks $(512-33+1)\times(384-33+1)=168960$	About 2000	Keypoint-blocks <4000
Feature dimension	64	32	8	128	2
The size of the sort matrix	730112×64	730112×32	168960×8	2000×128	$<4000\times2$
Offsets to be saved	46727168	23363584	1351680	256000	<8000
Locating accuracy	8×8	8×8	16×16	3×3	1×1
Computation complexity	$>O(32\times N\lg N)$	$O(32\times N\lg N)$	$O(8\times N\lg N)$	$O(\frac{N}{lg2}\lg N)$	$O(2\times N\lg N)$

Table 5. The correct detection rate under different $Threshold_1$, $Threshold_2$ and d.

		$d\times d$ (%)								
$Threshold_1$	$Threshold_2$	23 × 23	25 × 25	27 × 27	29 × 29	31 × 31	33 × 33	35 × 35	37 × 37	39 × 39
8.0	0.40	94.8	94.0	93.2	92.0	91.6	93.6	93.2	93.2	92.8
	0.45	94.8	94.0	92.8	92.0	92.8	94.0	92.8	93.6	93.2
	0.50	95.6	94.0	92.8	91.6	92.8	94.8	94.0	93.6	92.8
9.0	0.40	96.0	93.2	95.2	92.8	93.6	95.2	94.4	94.4	94.0
	0.45	95.6	94.0	95.6	92.4	90.0	96.4	94.8	94.8	94.8
	0.50	95.2	93.6	95.6	92.4	94.4	95.6	94.4	94.8	94.8
10.0	0.40	95.6	93.2	94.8	92.8	92.8	93.6	94.0	94.8	92.8
	0.45	95.6	93.6	94.4	92.8	90.0	96.8	93.6	94.8	93.6
	0.50	95.6	94.0	94.4	92.8	93.6	94.8	94.4	94.4	94.0
11.0	0.40	96.8	93.6	96.0	93.2	93.6	95.6	96.0	95.6	94.4
	0.45	96.0	94.0	96.0	92.8	92.8	95.6	95.2	95.6	95.2
	0.50	95.2	94.0	96.8	92.8	95.6	96.0	95.2	96.0	95.6

As can be seen from Table 4, in the proposed method, we use nonoverlapping image blocked mode, thus the number of blocks is less than that of Refs. 6–8. Our feature dimension, the size of the sort matrix, and the offsets to be saved are all least comparing with the compared methods. The less these numerical metrics, the higher is the efficiency of the algorithm. The locating accuracy is the minimum detectable unit, and it can be used to measure detection accuracy rate of the algorithm. The locating accuracy of Refs. 6 and 7 is 8×8, means that the minimum detectable region is 8×8 image block. It is also means the minimum size of error detection is 8×8. Similar, for Refs. 8 and 24, the minimum detectable size and the minimum error detection size are 16×16 and 3×3, respectively. However, the proposed method reaches pixel level accuracy, and the minimum detectable size and the minimum error detection size are all 1 pixel. It is superior to compared methods. Moreover, the computation complexity of the proposed method is lower than the compared methods. In conclusion, the comprehensive performance of the proposed method is satisfactory.

3.6. *Parameters setting*

The parameters used in our experiments are obtained from experimental statistics value. Through testing a large number of images, we found that the Threshold_1 and Threshold_2 used in our method in Sec. 2.3.1 were always in a certain range. In experiment, we tested 2000 forged images, and the sizes of tested images were 1024×768. We tested the correct detection rate (detection passing rate) under different thresholds. The correct detection rate is a rate of tampered image is correctly identified, and it is formally defined as follows:

$$\text{Correct detection rate} = \frac{\text{Number of tampered images are detected as tampered}}{\text{Total number of tested images}} \times 100\%. \tag{18}$$

Table 5 shows the tested results. As can be seen from Table 5, when $d = 33$, Threshold_1 is in 9.0 to 11.0, and Threshold_2 is in 0.45–0.50, the correct detection rate (detection passing rate) reaches higher value, so we set $d = 33$, $\text{Threshold}_1 = 10.0$ and $\text{Threshold}_2 = 0.45$.

4. Conclusion

Since copy-move attack is the most common image forgery method, the detection technology focus on copy-move attacks has attracted wide interests in recent years. In this paper, we present a novel image passive forensics method that can detect copy-move forgeries and duplicated image regions. The proposed method combines block matching technology and feature point matching technology. It has higher computation efficiency and can accurately detect duplicated regions of high

resolution images. The defect is it cannot detect tampered image with large angle rotation, therefore, there are a lot of works needed to be done in future.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant No. 61075007; the Natural Science Foundation of Shaanxi Province of China under Grant No. 2015JM6262.

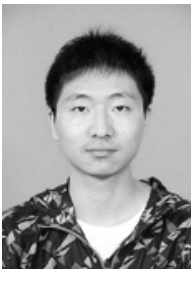
References

1. H. Farid, Exposing digital forgeries in scientific images, in *ACM Multimedia and Security Workshop* (2006).
2. J. Lukas, J. Fridrich and M. Goljan, Detecting digital image forgeries using sensor pattern noise, in *SPIE Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII Proceedings*, San Jose, California, USA (2006), pp. 362–272.
3. H. T. Sencar and N. Memon, *Overview of State-of-the-Art in Digital Image Forensics*, Part of Indian Statistical Institute Platinum Jubilee Monograph series titled 'Statistical Science and Interdisciplinary Research (World Scientific Press, 2008).
4. T. Ng, S. Chang, C. Lin and Q. Sun, Passive-blind image forensics, in *Multimedia Security Technologies for Digital Rights*, Chapter 15, eds. W. Zeng, H. Yu and C.-Y. Lin (Elsvier, 2006), pp. 383–412.
5. X. H. Li, Y. Q. Zhao, M. Liao, F. Y. Shih and Y. Q. Shi, Detection of tampered region for JPEG images by using mode based first digit features, *EURASIP J. Adv. Signal Process.* **190** (2012) 1–22.
6. J. Fridrich, D. Soukal and J. Lukas, Detection of copy-move forgery in digital images, in *Proc. Digital Forensic Research Workshop*, August 2003.
7. A. C. Popescu and H. Farid, Exposing digital forgeries by detecting duplicated image regions, in *Computer Science*, Dartmouth College, Tech. Rep. TR2004-515 (2004).
8. G. H. Li, Q. Wu, D. Tu and S. J. Sun, A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD, in *Proc. 2007 IEEE ICME*, Beijing (2007), pp. 1750–1753.
9. S. Bayram, S. Husrev and N. Memon, An efficient and robust method for detecting copy-move forgery, in *Proc. 2009 IEEE Int. Conf. Acoustics, Speech and Signal Processing* (2009), pp. 1053–1056.
10. B. Mahdian and S. Saic, Detection of copy-move forgery using a method based on blur moment invariants, *For. Sci. Int.* **107** (2007) 180–189.
11. W. Luo, J. Huang and G. Qiu, Robust detection of region duplication forgery in digital image, in *Proc. 18th Int. Conf. Pattern Recognition*, Hongkong, China, Vol. 4 (2006), pp. 746–749.
12. S. Khan and A. Kulkami, A efficient method for detection of copy-move forgery using discrete wavelet transform, *Int. J. Comput. Sci. Eng.* **2**(4) (2010) 1801–1806.
13. F. Y. Shih and Y. Yuan, A comparison study on copy-cover image forgery detection, *The Open Artif. Intell. J.* **4** (2010) 49–54.
14. G. Lynch, F. Y. Shih and H.-Y. M. Liao, An efficient expanding block algorithm for image copy-move forgery detection, *Inform. Sci.* **239**(7) (2013) 253–265.
15. D. Lowe, Distinctive image features from scale-invariant keypoints, *Int. J. Comput. Vision* **60**(2) (2004) 91–110.

16. H. Huang, W. Guo and Y. Zhang, Detection of copy-move forgery in digital images using SIFT algorithm, in *Proc. IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, Wuhan, China (2008).
17. I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo and G. Serra, Geometric tampering estimation by means of a SIFT-based forensic analysis, in *Proc. ICASSP*, Dallas, TX (2010).
18. I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo and G. Serra, A SIFT-based forensic method for copy move attack detection and transformation recovery, *IEEE Trans. Inform. Forensics Security* **6**(3) (2011) 1099–1110.
19. 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 Process. Image Commun.* **28**(5) (2013) 659.
20. R. Toldo and A. Fusiello, Robust multiple structures estimation with J-Linkage, in *Proc. ECCV*, Marseille, France (2008).
21. X. Pan and S. Lyu, Region duplication detection using image feature matching, *IEEE Trans. Inform. Forensics Security* **5**(4) (2010) 857–867.
22. P. Kakar and N. Sudha, Exposing postprocessed copy-paste forgeries through transform-invariant features, *IEEE Trans. Forensics Inform. Security* **7**(3) (2012) 1018–1028.
23. J. Beis and D. Lowe, Shape indexing using approximate nearest neighbor search in high-dimensional spaces, in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, IEEE Comput. Soc. (1997), pp. 1000–1006.
24. J. Li and S. Chao, Image copy-move forgery detecting based on local invariant feature, *J. Multimedia* **7**(1) (2012) 90–97.
25. L. Chen, W. Lu, J. Ni, W. Sun and J. Huang, Region duplication detection based on Harris corner points and step sector statistics, *J. Vis. Commun. Image Represent.* **24**(3) (2013) 244–254.
26. B. Xu, J. Wang, G. Liu and Y. Dai, Image copy-move forgery detection based on SURF, in *Int. Conf. Multimedia Information Networking and Security* (2010).
27. B. L. Shivakumar and S. Baboo, Detection of region duplication forgery in digital images using SURF, *Int. J. Comput. Sci. Iss.* **8** (2011) 199–205.
28. Q. Wu, S. Wang, G. Liu and Y. Dai, Log-polar based scheme for revealing duplicated regions in digital images, in *IEEE Signal Process. Lett.* **18**(9) (2011) 559–562.
29. B. S. Sergio and A. K. Nandi, Exposing duplicated regions affected by reflection rotation and scaling, in *IEEE Int. Conf. Acoustics, Speech and Signal Processing* (2011), pp. 1880–1883.
30. T. Zhang and R. Wang, Copy-move forgery detection based on SVD in digital images, in *2nd Int. Congress on Image and Signal Processing (CISP '09)*, Tianjin (2009), pp. 1–5.
31. C. Harris and M. Stephens, A combined corner and edge detector, in *Proc. Alvey Vision Conference*, University of Manchester (1988), pp. 147–151.

Xiaofeng Wang received her B.S. degree in Applied Mathematics from the Tianjin University, Tianjin, China, and both her M.S. degree in Mathematics and her Ph.D. in Mechanical and Electronic Engineering from the Xi'an University of Technology, Xi'an, China. In 2007, she joined the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, where she was a post-doctoral researcher until 2010. In 2012, she joined the Grasp Lab, the University of Pennsylvania, where she was a visiting scholar until 2013. She is currently a professor with the Department of Mathematics, Xi'an University of Technology, China. Her current research interests include multimedia forensics and security, image processing, steganography and steganalysis.

Guanghui He received his B.S. degree in Mathematics from Handan University, Hebei, China, in 2010, and his M.S. degree in Mathematics from the Xi'an University of Technology, Xi'an, China, in 2013. His research interests include multimedia forensics and security, and images processing.



Chao Tang received his B.S. degree in Mathematics from Xi'an University of Technology, Xi'an, China, in 2013. Currently he is studying for an M.S. degree in Mathematics, Xi'an University of Technology, Xi'an, China. His research interests include multimedia forensics and security, and images processing.

Yali Han received her B.S. degree in Mathematics from Xi'an University of Technology, Xi'an, China, in 2012, and her M.S. degree in Mathematics from the Xi'an University of Technology, Xi'an, China, in 2015. Her research interests include multimedia forensics and security, and images processing.

Shangping Wang received his B.S. degree in Mathematics in 1982 from Xi'an University of Technology, Xi'an, China. He received his M.S. degree in Applied Mathematics in 1989 from Xi'an Jiangtong University, Xi'an, China, and earned his Ph.D. in Cryptology in 2003 from Xidian University, Xi'an, China. Currently, he is a professor at Xi'an University of Technology. His current research interests are cryptography and information security.