Forensic Science International 214 (2012) 33–43

Contents lists available at ScienceDirect

Forensic Science International

journal homepage: www.elsevier.com/locate/forsciint

A robust detection algorithm for copy-move forgery in digital images

Yanjun Cao [a,*], Tiegang Gao [b], Li Fan [a], Qunting Yang [a]

[a] College of Information Technical Science, Nankai University, Tianjin 300071, China
[b] College of Software, Information Security Technology Lab, Nankai University, Tianjin 300071, China

ARTICLE INFO

Article history:
Received 21 December 2010
Received in revised form 9 June 2011
Accepted 6 July 2011
Available online 2 August 2011

Keywords:
Digital forensic
Copy-move forgery
Circle block
Region duplication detection

ABSTRACT

With the availability of the powerful editing software and sophisticated digital cameras, region duplication is becoming more and more popular in image manipulation where part of an image is pasted to another location to conceal undesirable objects. Most existing techniques to detect such tampering are mainly at the cost of higher computational complexity. In this paper, we present an efficient and robust approach to detect such specific artifact. Firstly, the original image is divided into fixed-size blocks, and discrete cosine transform (DCT) is applied to each block, thus, the DCT coefficients represent each block. Secondly, each cosine transformed block is represented by a circle block and four features are extracted to reduce the dimension of each block. Finally, the feature vectors are lexicographically sorted, and duplicated image blocks will be matched by a preset threshold value. In order to make the algorithm more robust, some parameters are proposed to remove the wrong similar blocks. Experiment results show that our proposed scheme is not only robust to multiple copy-move forgery, but also to blurring or nosing adding and with low computational complexity.

1. Introduction

Nowadays, with the popularity of low-cost and high-resolution digital cameras, digital media is playing a more and more important role in our daily life, however, due to the sophisticated editing software (for example, Photoshop, 3D Max), digital images can be easily manipulated and altered without leaving visible clues, thus, it poses a serious social problem as to how much of their content can be trusted, whether it is authentic or tampered especially as a witness in a courtroom, insurance claims and scientific fraud. According to some statistics [1], in one journal, as many as 20% of accepted manuscripts contain figures with inappropriate manipulations, and 1% of which with fraudulent manipulations. As a result, when the counterfeit images are used for vicious purpose, it may result in inestimable lose. To combat this problem, digital image forensics has emerged as a new research filed to reveal digital tampering in images.

There are several types of tampering, however, concealing some objects from natural images is a common form of digital image tampering, known as copy-move forgery (CMF). An example of CMF is shown in Fig. 1. Some researchers have developed techniques to deal with CMF, they all use square blocks for matching purpose. Fridrich [2] used DCT-based features instead of exhaustive search to detect region duplication, which is more effective, but their method is sensitive to variations in duplicated regions owing to additive noise. Later, Huang et al. [3] improved the performance by reducing the feature vector in dimension, however, they failed to consider the multiple copy-move forgery. In [4], Popescu proposed a new method by adopting the PCA-based feature, which can endure additive noise, but the detection accuracy is low. Luo [5] proposed color features as well as block intensity ratio to show the robustness of their method. Bayram et al. applied Fourier-Mellin transform (FMT) to each block [6], FMT values are finally projected to one dimension to form the feature vector. [7] used a method based on blur moment invariants to locate the forgery regions, and [8] took the advantage of the SIFT features to detect the duplication regions and their experiments show the robustness of their approach. Yet, the methods mentioned above have higher computational complexity, since the quantized square blocks are directly used for matching, that the dimension of feature vector is higher, as a consequence, affecting the efficiency of detection, especially when dealing with high-resolution digital images.

In this paper, we propose a robust and efficient detection approach based on improved DCT. Compared with other methods, the main advantages of our method can be summarized as:

1. The dimension of the feature vector is lower;
2. It is robust to various attacks, such as: multiple copy-move forgery, Gaussian blurring, and noise contamination;

* Corresponding author. Tel.: +86 22 23504956.
E-mail addresses: caoyanjun528411@gmail.com (Y. Cao), gaotiegang@gmail.com (T. Gao), fanlihlm@gmail.com (L. Fan), quntingyang@gmail.com (Q. Yang).

0379-0738/$ – see front matter © 2011 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.forsciint.2011.07.015

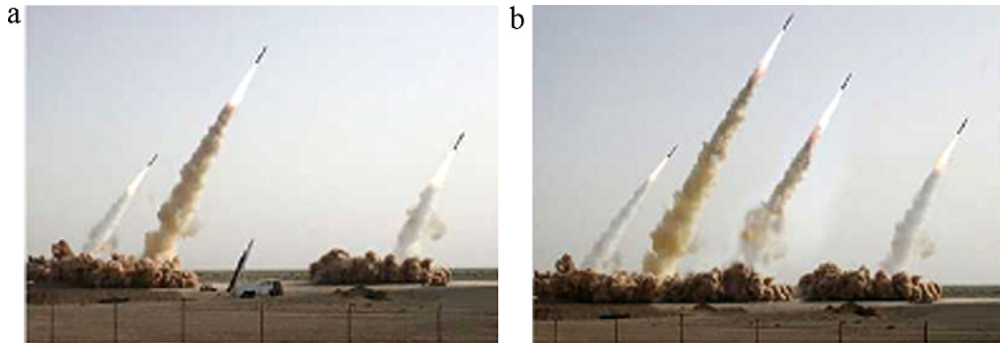

Fig. 1. An example of CMF [8]: (a) the original image with three missiles and (b) the forged image with four missiles.

3. It has lower computational complexity.

The rest of the paper is organized as follows: in Section 2 the proposed approach is described. Section 3 gives some experimental results and gives the corresponding analysis. Conclusion is drawn in Section 4.

2. The proposed approach

Generally speaking, a nature image is unlikely to have two large similar regions (except that there are two large regions, such as blue sky in an image), [5] also testify that. The task of finding copy-move forgery is to find large similar regions in an image. Since the duplicated regions are unknown both in size and shape, if we compare every possible pairs pixel by pixel, the computational complexity will be very higher, none can endure that. Obviously, it is more practical to divide the suspicious image into blocks for detecting the duplicated regions.

In order to take an efficient detection, some appropriate and robust features must be extracted from the blocks, therefore, a good features extraction can not only represent the whole blocks, but also reduce the dimension of feature vector, and what's more, make the detection algorithm has lower computational complexity.

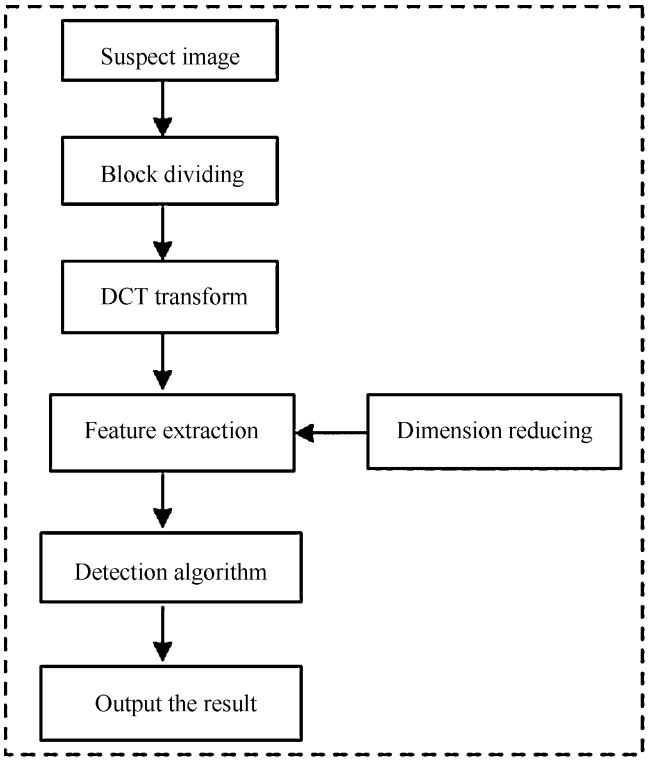

Fig. 2. Architecture of the detection algorithm.

2.1. Methods

According to the above discussion, the whole detection framework is given as follows:

(1) Dividing the suspicious image into fixed-size blocks.
(2) DCT is applied to each block to generate the quantized coefficients.
(3) Representing each quantized block by a circle block and extracting appropriate features from each circle block.
(4) Searching similar block pairs.
(5) Finding correct blocks and output them.

The entire architecture of the proposed algorithm can also be seen from Fig. 2.

2.2. Implementation details

In our algorithm, we first divided the original image into fixed-size blocks and then detect the similarity of these blocks and finally output the possible duplicated regions. Details are as follows:

Step 1: Assuming a $M \times N$ grayscale image I (if the image is color, we can use the standard formula: $I = 0.228R + 0.587G + 0.114B$ to turn it to grayscale), we first split it into overlapping blocks of $B \times B$ pixels, that is, the adjacent blocks only have one different row or column. Each block is denoted as B_{ij}, where i and j indicates the starting pointing of the block's row and column, respectively.

$$B_{ij}(x,y) = f(x+j, y+i), \tag{1}$$

where $x, y \in \{0, \ldots, B-1\}$, $i \in \{1, \ldots, M-B+1\}$, and $j \in \{1, \ldots, N-B+1\}$

Hence, we are able to obtain N_{blocks} of overlapped sub-blocks from suspicious image.

$$N_{blocks} = (M-B+1) \times (N-B+1) \tag{2}$$

Step 2: Let $N_{\text{blocks}} = (M-B+1) \times (N-B+1)$, for each block $B_i (i = 1, 2, 3 \ldots N_{\text{blocks}})$, DCT is applied. After that a DCT coefficients matrix with the same size as the block is exploited, which can represent the corresponding block.**Step 3:** Since each block is represented by the DCT coefficients, here, we assume the size of the block B_i is 8×8, the size of the coefficients matrix is also 8×8, obviously, there are 64 elements in the matrix. As it is the nature of DCT that the energy only focuses on the low frequency coefficients, that is, not all of the elements are equal important, the low frequency coefficients play the important role.

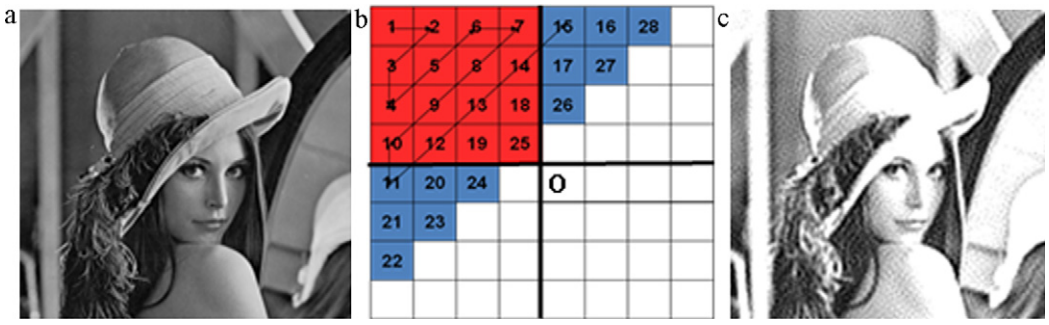

Fig. 3. (a) the Lena image (b) Zigzag order scanning (c) the reconstruction image of Lena by using 1/4 DCT coefficients.

For this, we make an illustration, we use an image of Lena with the size of 256 pixels × 256 pixels. Fig. 3(a) is Lena, then the discrete cosine transform is applied to Fig. 3(a), after that, we extract the low frequency DCT coefficients of Fig. 3(a) in a zigzag order, Fig. 3(b) is an example of zigzag order, the red area is the low frequency part, which occupies the 1/4 energy of the entire DCT coefficients. Fig. 3(c) is the reconstruction image of Lena after extracting the 1/4 DCT coefficients of Fig. 3(a) in a zigzag order. Through the analysis of Fig. 3, if the image block undergoes DCT transform, we can use four-part energy to represent the whole image while without losing any important information. For this basic motivation, we use a circle block to represent the coefficients matrix and divide it into four parts: C_1,C_2,C_3,C_4, as can be seen from Fig. 4.

Denote r as the radius of the circle block, so we can get the area:

$$c\_\text{area} = pi \times r^2$$

and

$$m\_\text{area} = 4r^2$$

where $c\_$ area, $m\_$ area is the area of the circle block and matrix, respectively. Let

$$p\_\text{ratio} = \frac{c\_\text{area}}{m\_\text{area}} \tag{3}$$

From Eq. (3), we can get $p\_\text{ratio} = pi \times r^2/4r^2 \approx 0.79$, which implies that the circle block can represent most of the elements of the matrix, and discards only a few of them, in addition, Fig. 3 shows the nature of DCT that the energy focuses on the low frequency, using a circle block instead of a square block does not affect the detection efficient, on the contrary, it can decrease the computational complexity.

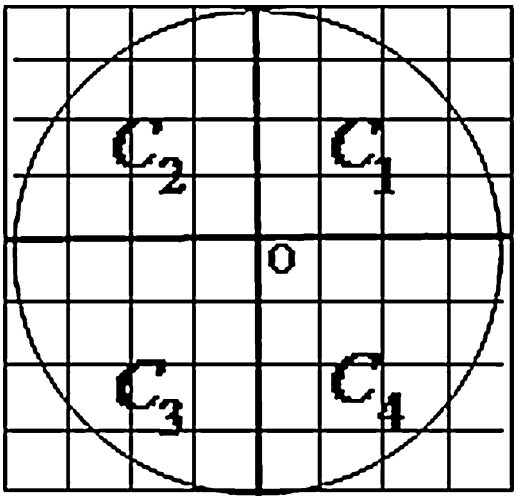

Fig. 4. Feature extraction.

From the above analysis, in order to obtain the matching features, we denote v_1, v_2, v_3, v_4 as the feature of c_1, c_2, c_3, c_4 respectively. We can get $v_i (i = 1, 2, 3, 4)$ through Eq. (4).

$$v_i = \frac{\sum f(x,y)}{c\_\text{area}_i}, \quad (f(x,y) \in c\_\text{area}_i, \ \ i = 1,2,3,4) \tag{4}$$

where v_i is the mean of the coefficients value, corresponding to each c_i. Since each c_i is represented by different DCT coefficients and which can represent the energy of the image, each v_i can also be taken as a quantized by c_area_i. After that, four features are gotten, which can be combined to a feature vector with the size of 1×4, denote as:

$$V = [v_1, v_2, v_3, v_4] \tag{5}$$

so a 8×8 matrix is represented by a 1×4 feature vector, compared with [2–4], which is a 1×64, 1×16, 1×32 feature vector, the dimension of ours is lower.

These features will not change a lot after some common post-processing operations. Taking the additive white Gaussian noise operation for example, if an image is contaminated by noise adding operation, then the pixel value will be changed, for each pixel, we define $f_{x,y} = \lfloor f_{x,y} \rfloor + \xi_{x,y} (0 < \xi_{x,y} < 1)$, where $f_{x,y}$ is the corresponding pixel value that contaminated by signal noise, $\lfloor f_{x,y} \rfloor$ is the nearest value less than or equal to the original pixel value, $\xi_{x,y}$ is the random signal noise for each pixel, we assume these signal noise are IID (independent identically distributed) and with mean 0, then each noisy sub-block $B'_{ij} = B_{ij} + \xi_{x,y}$, by using Eq. (4), we can get $V'_1 = V_1 + \xi'_{x,y}$, since $E(\xi'_{x,y}) = 0$, $D(\xi'_{x,y}) = \sum_{j=1}^{B^2} \xi'_{x,y}/B^2$, generally $\sum_{j=1}^{B^2} \varepsilon'^2_{x,y} \ll B^2 (0 < \varepsilon'_{x,y} < 1)$, we can get $V'_1 \approx V_1$, similarly for $V'_2 \approx V_2$, $V' \approx V_3$ and $V'_4 \approx V_4$. For Gaussian blurring operation, the Gaussian blur filter only affects some high frequency components of each sub-block but changes the low frequency components a little.

In order to show the robustness of the feature vector, we randomly select a 8×8 block from the image of Baboon, and perform some post-processing operations with different parameters, which can be seen from Table 1. Let V_{original} and $V_{\text{post\_processing}}$ as two feature vector, after some post-processing, we compute the correlation between them, if the result is close to 1, which implies the feature vector is robustness and the invariance is more stable.

From Table 1, we can see that, after some post-processing, the correlation coefficient between V_{original} and $V_{\text{post\_processing}}$ is close to 1 and the features we extracted are robust, V can be used as the feature vector, thus, the dimension reduction is achieved.

Table 1
The robustness of the feature vector.

Feature vector	$V_{original}$	$V_{post\_processing}$			
		AWGN (SNR = 25 db)	AWGN (SNR = 50 db)	Gaussian blurring ($w = 3, \sigma = 1$)	Gaussian blurring ($w = 5, \sigma = 0.5$)
v_1	−0.3240	−0.6580	−0.4648	−0.3229	−0.3241
v_2	−0.1761	−0.3016	−0.2342	−0.1762	−0.1761
v_3	−0.0089	−0.0181	−0.0186	−0.0089	−0.0090
v_4	0.0370	0.0063	0.0277	0.0363	0.0370
Correlation coefficient		0.9904	0.9981	1.0	1.0

Step 4: The feature vectors extracted from Step 3 are arranged to a matrix, denote as A with the size of $(M - B + 1)(N - B + 1) \times 4$.

$$A = \begin{bmatrix} V_1 \\ \vdots \\ V_{(M-B+1)(N-B+1)} \end{bmatrix} \tag{6}$$

The A is then lexicographically sorted, meantime, record the left corner's coordinate of each block which represented by a circle block. Since each element of A is a vector, the sorted set is defined as $\hat{A}$. Based on $\hat{A}$, the Euclidean distance $m\_match(\hat{A}_i, \hat{A}_{i+j})$ between adjacent pairs of $\hat{A}$ is calculated. If the distance is smaller than a preset threshold $D_{similar}$, then we initialize a black map P with the size $M \times N$ and consider the inquired blocks as a pair of candidate for the forgery.

$$\hat{A}_i = (\hat{v}_i^1, \hat{v}_i^2, \hat{v}_i^3, \hat{v}_i^4),$$

$$\hat{A}_{i+j} = (\hat{v}_{i+j}^1, \hat{v}_{i+j}^2, \hat{v}_{i+j}^3, \hat{v}_{i+j}^4),$$

$$m\_match(\hat{A}_i, \hat{A}_{i+j}) = \sqrt{\sum_{k=1}^{4} (v_i^k - v_{i+j}^k)^2} < D_{similar} \tag{7}$$

In addition, due to the fact that the neighboring blocks may have the similar feature vector, we calculate the actual distance between two similar blocks as follows:

$$m\_distance(V_i, V_{i+j}) = \sqrt{(x_i - x_{i+j})^2 + (y_i - y_{i+j})^2} > N_d \tag{8}$$

here (x, y) is the circle center of the corresponding block, we use Eqs. (7) and (8) to determine whether the blocks are duplicated or not.

In sum, in order to make the detection, we set four thresholds: the sliding window B, similarity threshold $D_{similar}$, distance threshold N_d, and N_{number} which controls the amount of neighboring feature vectors, only if the test satisfies the following condition:

$$m\_match(V_i, V_j) < D_{similar} \,\&\, m\_distance(V_i, V_j) > N_d \tag{9}$$

where $j \in [i - N_{number}, i + N_{number}]$, we mark a color map for the original block and another color map for the duplicated block.

Step 5: Morphologic operations are applied to P to fill the holes in the marked regions and remove the isolated regions, then output the final result.

2.3. Analysis of computational complexity

This section we analyze the computational complexity. Firstly, N_{blocks} sub-blocks must be divided from the suspicious image ($M \times N$ in dimension), thus, we need about $O(N_{blocks})$ time to obtain all the sub-blocks. After that, a two-dimensional discrete cosine transform is applied to each sub-block, we assume each sub-block is b pixels ($\sqrt{b} \times \sqrt{b}$ pixels in dimension), then from the analysis of Section 2.2, about $O(\sqrt{b} \times \sqrt{b})$ time is needed to compute a single sub-block. Obviously, we need about

$$O(N_{blocks} \times \sqrt{b} \times \sqrt{b})$$

time to obtain all the DCT coefficients of each sub-block. Then in order to make a representation for each sub-block, four features are extracted from each transformed sub-block, since for each feature extracting, each pixel that inside the circle block is scanned, and this will take about

$$O(4 \times \sqrt{b} \times \sqrt{b})$$

time to calculate the four features. Hence, the entire time for extracting all the features of each sub-block is about $O(N_{blocks} \times 4 \times \sqrt{b} \times \sqrt{b})$. Finally, a lexicographical sorting is involved for matrix A ($N_{blocks} \times 4$ in dimension), it roughly takes

$$O(4 \times N_{blocks} \times \log N_{blocks}).$$

Therefore, the total computational complexity is approximate to

$$O(N_{blocks}) + O(N_{blocks} \times \sqrt{b} \times \sqrt{b}) + O(N_{blocks} \times 4 \times \sqrt{b} \times \sqrt{b}) + O(4 \times N_{blocks} \times \log N_{blocks})$$

But we should also notice that, since N_{blocks} is obtained by Eq. (2), for a given image with $M \times N$ pixels, as the size of the image grows, the total number of sub-block increase with a $O(M \times N)$ complexity, which is another reason affecting the whole computational complexity.

Through the above analysis, a key problem in the detection algorithm is the computational complexity, which is caused by the amount of the matching blocks and the dimension of the feature vector. There are some researchers use different methods to reduce the computational complexity, for example [2–4], use DCT-based, Improved DCT-based, and PCA method respectively.

In this paper, our algorithm focuses on the dimension of feature vector. We use a circle block to represent each block which is quantized by DCT, and then four features are extracted, compared with [2–4], the amount of the dividing blocks are same, however, the feature vector's dimension of ours is lower, which implies our method has a lower computational complexity and Table 2 also makes a comparison with them.

3. Experimental results and analysis

We use Photoshop 8.0 to modify the images and all the tests are carried out on the platform with a 2.59 GHz, AMD processor and

Table 2
Computational complexity comparison.

Literatures	Extraction method	Feature dimension
[2]	DCT	64
[4]	PCA	32
[3]	Improved DCT	16
Ours	Block representing	4

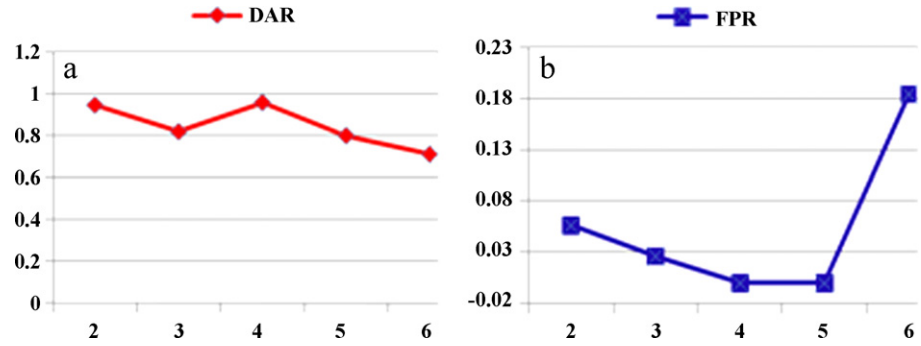

Fig. 5. Shown are the DAR/FPR performance under different circle radius ranging from 2 to 6 with 1increment. Each data point corresponds to an average over 200 images.

Matlab2009a. In our experiments, the tampered images are generated based on three datasets. The first dataset are gray images came from the dataset of DVMM lab at Columbia University with the size of 128 pixels × 128 pixels [9], the second dataset are several uncompressed color PNG images of size 768 pixels × 512 pixels released by the Kodak Corporation [10], both for research purpose. Besides, we collected some images of size 1600 pixels × 1000 pixels from the internet, they all have large resolution and with big uniform areas, these images formed the third dataset. By using the proposed method, for a gray and color image from dataset I, dataset II and dataset III, it takes about 1.5 s, 35 s and 2.9 min to locate the tampered areas. However, if we use some advanced programming languages (e.g., C, C++), our experiments will take less time.

3.1. Thresholds setting

In order to quantify the accuracy and robustness of our algorithm, we set Ψ_S is the copy region, $\tilde{\Psi}_S$ is the detected copy

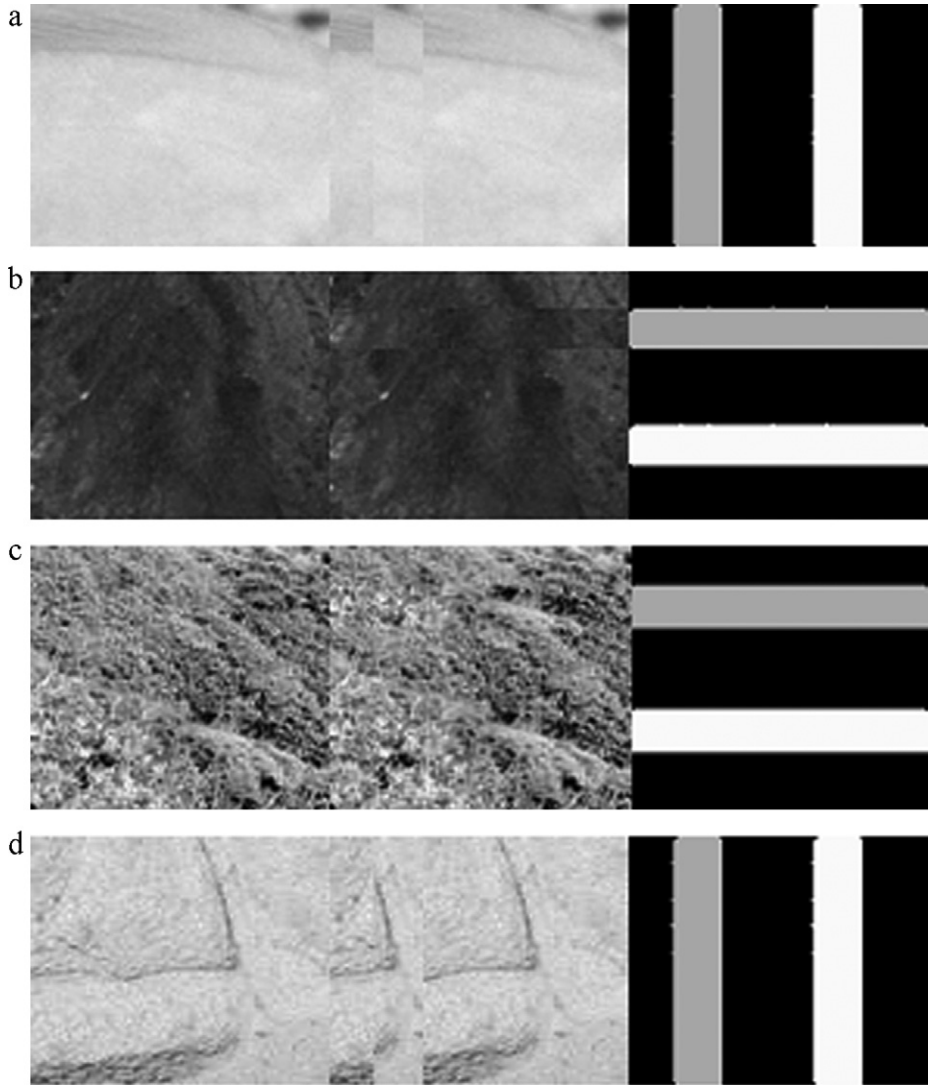

Fig. 6. Shown are the detection results (from left to right is the original image, tampered image, detection results).

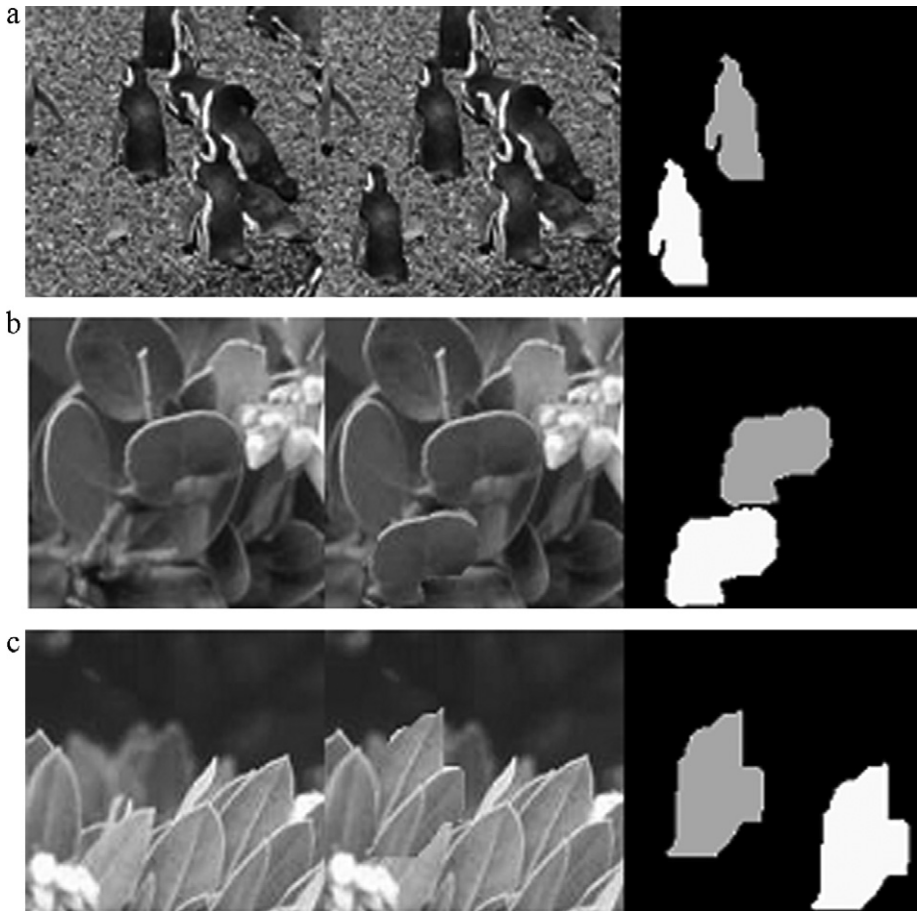

Fig. 7. Shown are the detection results for non-regular copy-move forgery.

region, Ψ_T and $\tilde{\Psi}_T$ is the tampered region and detected tampered region respectively. *DAR* is the detection accuracy rate, *FPR* is the false positive rate. Here, *DAR* and *FPR* are calculated as follows:

$$DAR = \frac{\left|\Psi_S \cap \tilde{\Psi}_S\right| + \left|\Psi_T \cap \tilde{\Psi}_T\right|}{|\Psi_S| + |\Psi_T|} \tag{10}$$

$$FPR = \frac{\left|\tilde{\Psi}_S - \Psi_S\right| + \left|\tilde{\Psi}_T - \Psi_T\right|}{\left|\tilde{\Psi}_S\right| + \left|\tilde{\Psi}_T\right|} \tag{11}$$

Since we use the overlapped sub-blocks and circle representation method for extracting the matching features, selecting the radius of the circle window is usually a tricky thing, in order to set the threshold parameters, we randomly choose about 200 images from the three datasets and then make a series of forgeries with the duplicated region size of 64 pixels × 64 pixels. After that, we use different circle radius ranging from 2 to 6, with 1 increment, then a set of values for B, D_{similar}, N_d, and N_{number} are gotten. Fig. 5 is the DAR/FPR curves under different circle radius, as can be seen from Fig. 5(a), with the increase of the circle radius, the DAR

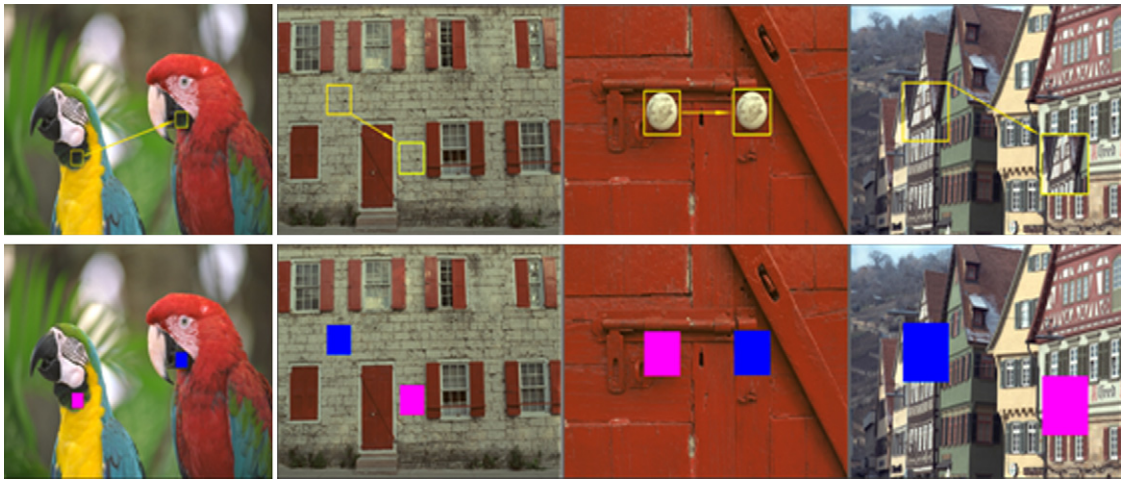

Fig. 8. . Shown on the top row are four images with duplicated region size of 32 pixels × 32 pixels, 64 pixels × 64 pixels, 96 pixels × 96 pixels, and 128 pixels × 128 pixels. Shown below are the detection results using our algorithm.

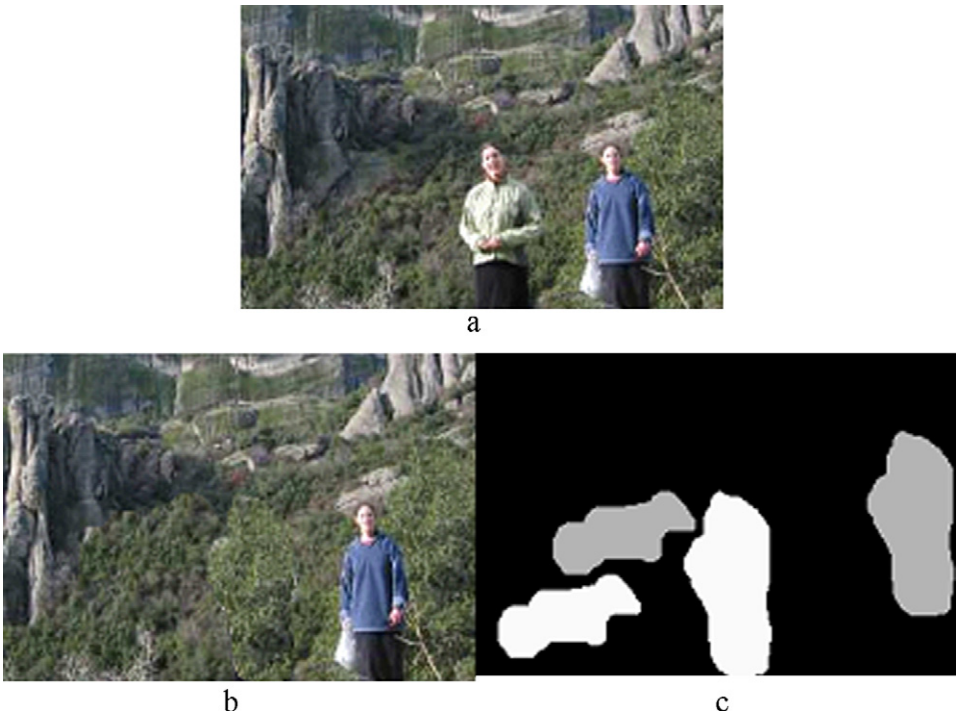

Fig. 9. Shown are the detection results for multiple copy-move forgery.

performance is prone to decrease on the whole, but there is an inflexion when $r = 4$, with the DAR performance approximating to 1 while has the lowest FPR. Therefore, in order to make a balance between the DAR and FPR performance, we set the circle radius $r = 4$, for color images, the optimal values turn out to be 8, 0.0015, 120 and 5 for B, D_{similar}, N_d, and N_{number}, respectively, for gray ones, the optimal values are set to 8, 0.0025, 25 and 5 for B, D_{similar}, N_d, and N_{number}, respectively.

3.2. Effectiveness testing

In order to test the effectiveness of the proposed method, for the first experiment, we choose some gray images having large and similar regions with the size of 128 pixels $\times$ 128 pixels from dataset I, the detection results can be seen from Fig. 6.

The images shown in Fig. 6 are the detection results without any post-processing operation, each set of tests, including three

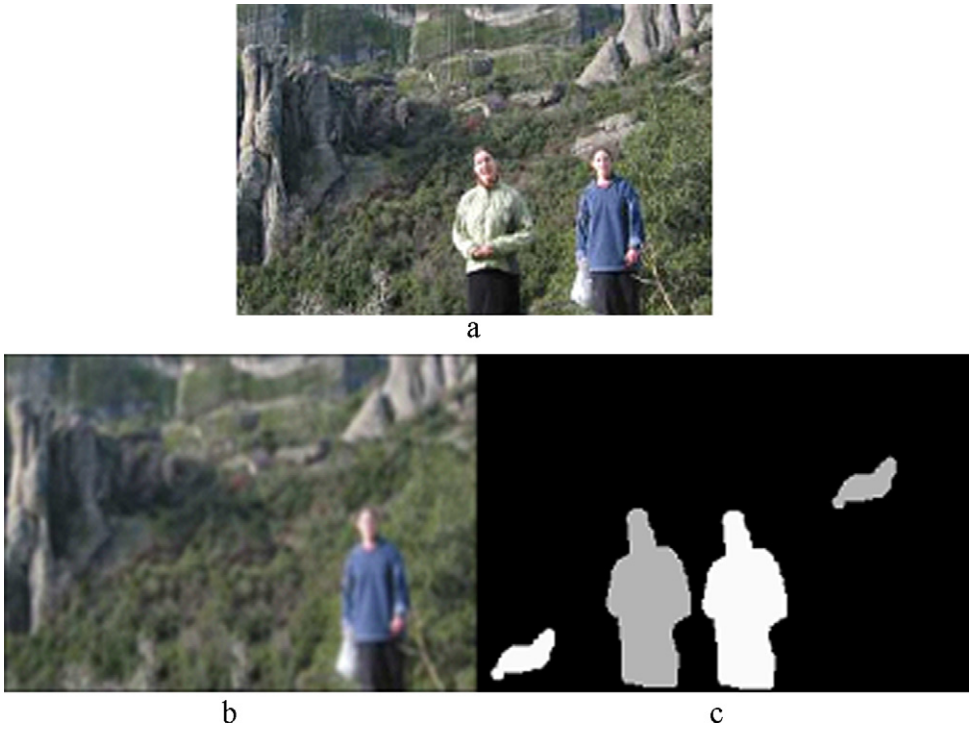

Fig. 10. Shown are the test results for multiple copy-move forgery under a mixed operation.

images: from left to right is the original image, tampered image, and detection result.

As can be seen from Fig. 6, because of the homogenous background, it is difficult to discern the doctored images, however, Fig. 6 shows that our algorithm can locate the tampered regions soundly, despite there are large and similar regions in the image. Another test is given in Fig. 7. We can see from Fig. 7, all the duplication regions are non-regular, the detection algorithm can find the tampered regions precisely.

The above tampered images are 128 pixels × pixels128 pixels, in the following test, we select some 768 pixels × 512 pixels images with complex texture background from dataset II, and randomly choose some different square regions for testing the effectiveness of our algorithm. Here, we use four different block sizes (32 pixels ×32 pixels, 64 pixels × 64 pixels, 96 pixels ×96 pixels, and 128 pixels ×128 pixels), corresponding to 0.26%, 1.04%, 2.34%, and 4.17% of total image area, respectively. In order to make a better visibility, we use different colors to mark the duplication regions.

Fig. 8 displays the detection result for the images with the tampered size of 32 × 32, 64 × 64, 96 × 96, and 128 × 128, respectively. The top row is the doctored images, with the yellow square indicates the copy-move region and the yellow arrow denotes the pasting location, the bottom is the detection results. In this example, no further manipulations were carried out on the tampered regions, as can be seen from Fig. 8, the tampered regions can be located perfectively, though each image has complex texture background.

3.3. Robustness and accuracy test

In this section, some robustness and accuracy analysis for the proposed method is given in detail. With the sophisticated image editing software, the tamper usually make their effort to create a plausible tampered image. Multiple copy-move as a mean of forgery has being gradually used for image manipulation, in an image, where there are several duplication regions. In our experiments, we take it into account, however

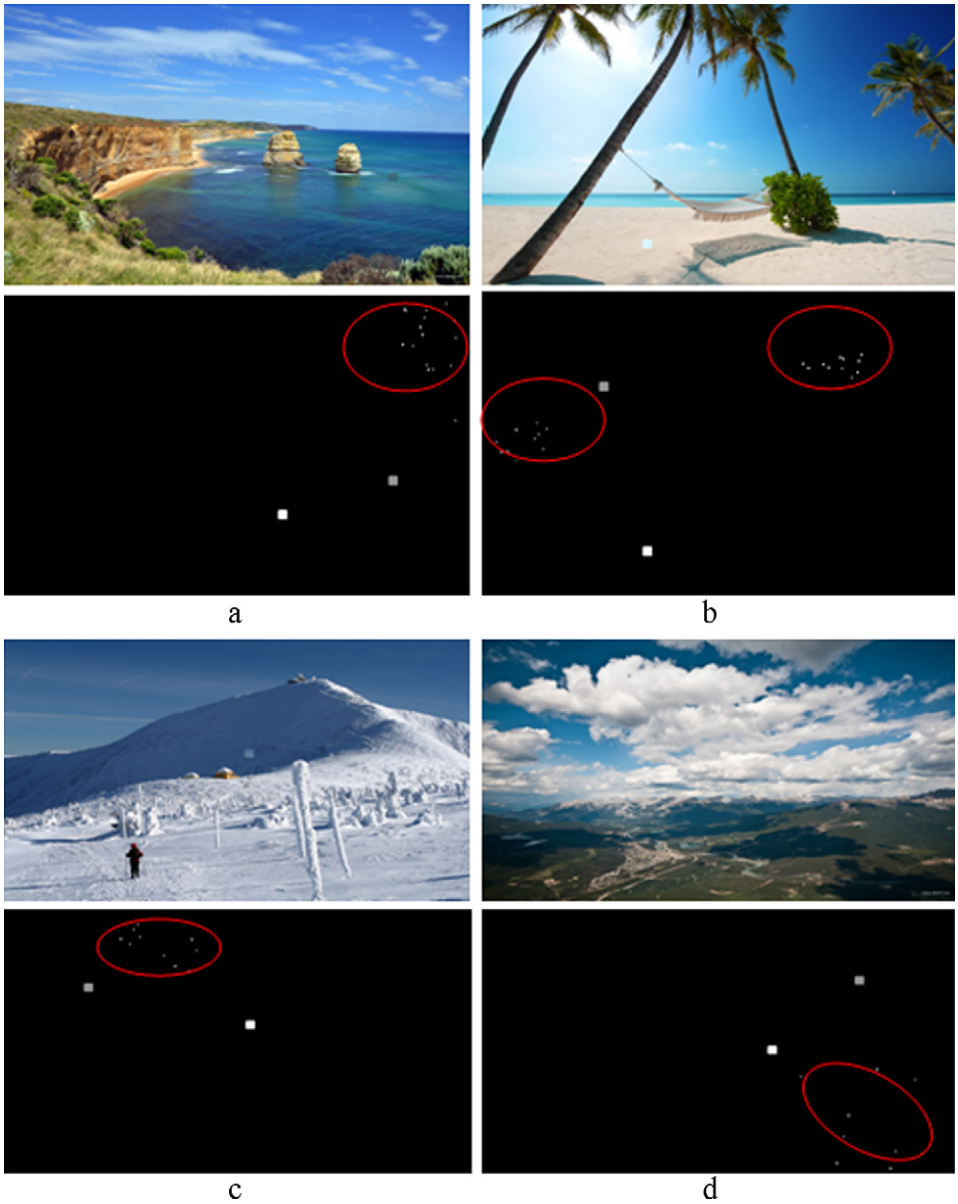

Fig. 11. Shown on the top row are four tampered images with duplicated region size of 32 pixels × 32 pixels. Shown below are the detection results using our algorithm.

[2–6] failed to consider such forgery. The test result is shown in Fig. 9.

In Fig. 9, Fig. 9(a) is the original image which has two people, with some small hills and shrubbery as the background. Fig. 9(b) is the tampered image, we copied two parts of the shrubbery to conceal one of the people, as can be seen from Fig. 9(b), the tampered image is seemly like a natural image, however, by using our algorithm, it detected two duplication regions in Fig. 9(b), as showing in Fig. 9(c).

In real life, some evil people often handle the tampered images with post-processing operation, such as noise adding, blurring or mixture operations, in order to achieve some purpose. Fig. 10 shows such situation. Fig. 10(a) is the original image, Fig. 10(b) is the tampered image with mixed operations, and Fig. 10(c) is the detection result. Firstly, the tampered image is blurred by a Gaussian blur filter with $w = 5, \sigma = 1$, and then the tampered image is corrupted by adding white Gaussian noise with SNR = 15 db. As can be seen from Fig. 10(b) which suffered the above processing, we can hardly discern the whole image, however, Fig. 10(c) shows us that our algorithm can locate the multiple duplication regions with a satisfactory degree, even though the image is processed by mixed operations. Literature [2–6] does not give such experiment.

The last thing I want to note is that, with the powerful of the digital devices, people can get a higher resolution image easily. Thus, it raises a problem, when only a small fraction of the image is tampered, it is usually hard to locate the duplication region, in addition, with the increase of the image size, the computational complexity is also a big challenge. To address these two issues, we select some images with the size of 1600 pixels × 1000 pixels from dataset III, they all have higher resolution and with big uniform areas. Fig. 11 is the detection result.

In Fig. 11, the top row are the tampered images, the bottom are the detection results using our algorithm. For each one, we only doctored a small fraction of the total image, with the duplicated region size is 32 pixels × 32 pixels, about 0.064% of the total image area. As can be seen from Fig. 11, all the tampered areas can be detected perfectively, but there are some mismatching blocks, we denote them by red circles. In Fig. 11(a), the false positive blocks are on the top-right corner, where the background is the uniform blue sky. In Fig. 11(b), there are two mismatching areas, just as the two red circles denote, it is ascribed to the big uniform areas, since the matching features that extracted from them are approximately the same. For Fig. 11(c) and (d), they both have one mismatching area, the reason is similar as Fig. 11(a) and (b).

With regard the computational complexity, for each 1600 pixels × 1000 pixels image, in our experiments, it took about 2.9 min to complete the detection, we think the performance is relatively exciting. Though the detection results may not be very satisfactory in some cases, we should note that, at present, there is no omnipotent tool that can against any situation, our approach is no exception.

Furthermore, in order to quantify the sensitivity and efficiency of our algorithm to image degradations, we again select about 200 images with the size of 128 pixels × 128 pixels and 768 pixels × 512 pixels from dataset I and II. Then randomly copy a square region and paste it to a non-overlapping position. The tampered images are then distorted by additive white Gaussian noise and Gaussian blurring operations. In our test, the sizes of the square regions are of 16 × 16, 24 × 24, 32 × 32, and 48 × 48. The average *DAR*, *FPR* performance over 200 images are shown in Fig. 12.

As shown in Fig. 12, the whole DAR/FPR performance of our approach is relatively robust to these operations. In most of the cases, about more than 80% of the area in copy-move regions can be identified with less than 13% of FPR, even though the image with poor quality (SNR = 10 or $w = 5, \sigma = 3$) and small duplication regions (16 pixels × 16 pixels). In general, the performance is prone to decrease when the image quality is poor, however, Fig. 12 shows us that, our algorithm can detect more than 80% of the tampered regions, in reality, based on human interpretation, it can give a reliable clue for the inspector to verify the authenticity of an image.

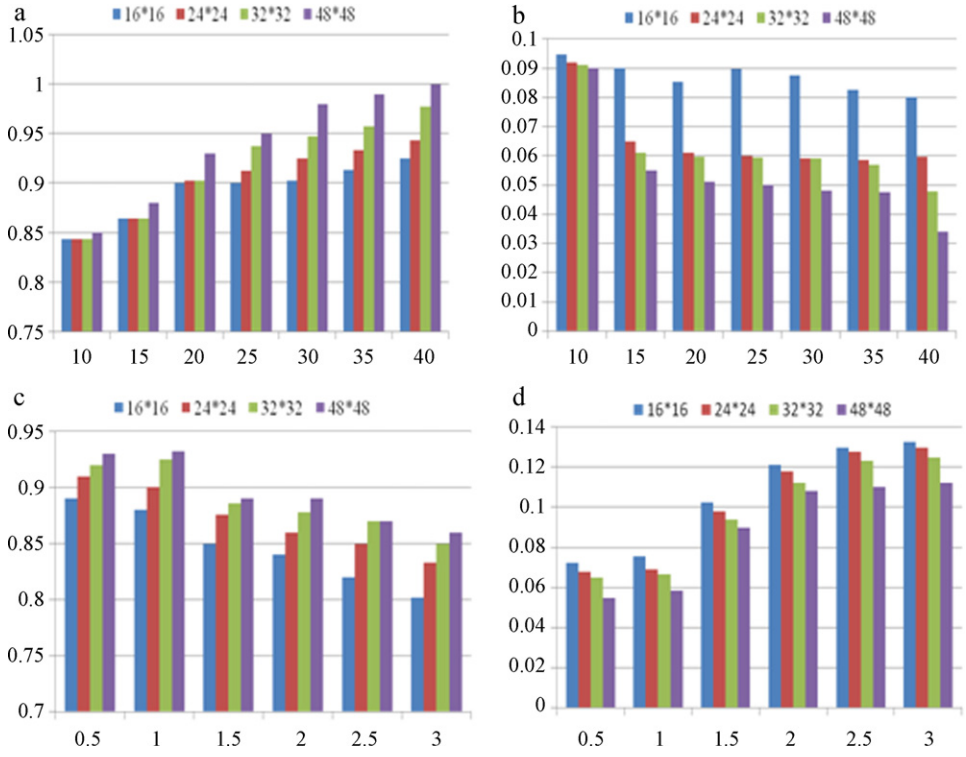

Fig. 12. Shown is the average detection accuracy. (a)–(b) DAR/FPR performance with SNR = 10 db, 15 db, 20 db, 25 db, 30 db, 35 db,40 db and (c)–(d) DAR/FPR performance with Gaussian blurring ($w = 5$, $\sigma = 0.5$, 1, 1.5, 2, 2.5, 3). Each data point corresponds to an average over 200 images.

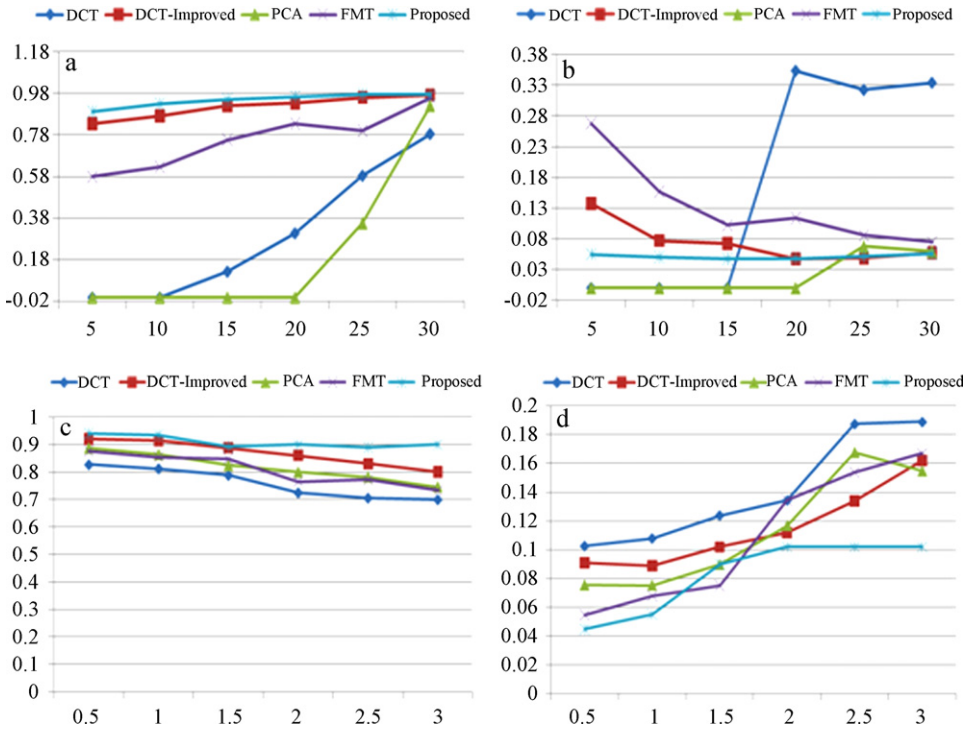

Fig. 13. DAR/FPR curves for DCT, DCT-improved, PCA, FMT, and Proposed methods when the duplicated region is 64 pixels × 64 pixels. (a)–(b) with different SNR levels (SNR = 5 db, 10 db, 15 db, 20 db, 25 db, 30 db), and (c)–(d) with Gaussian blurring ($w = 5$, $\sigma = 0.5, 1, 1.5, 2, 2.5, 3$).

In the last experiment, we compared our method with other approaches: DCT-based [2], DCT-improved [3], PCA-based [4], and FMT-based [6]. Before the comparison, we created about 100 tampered images, for each image, we randomly copied a region of size 64 pixels × 64 pixels and pasted it to another non-overlapping location. After that, the forged images underwent several post-processing operations, in this paper, we only consider two cases: additive white Gaussian noise, and Gaussian blurring, since these two operations are commonly used in real life.

The overall average performance comparison over 100 tampered images is shown in Fig. 13. In case of noise adding, Fig. 13(a)–(b), where the tampered images are contaminated with additive white Gaussian noise (SNR = 5,10,15,20,25,30 db). With the increase of SNR levels, the DAR also increased for all methods. Observation from the DAR/FPR curves show that, the PCA-based method has the lowest DAR than other methods, followed by DCT-based and FMT-based, especially when the SNR is less than 20 db, the DAR of PCA-based is approximate to zero. Our method achieves higher DAR than other methods, even though with lower SNR level (SNR = 5 db). For FPR, as can be seen from Fig. 13(b), when the SNR is less than 20 db, PCA-based method has lower FPR, approximates to zero, since in such case, PCA-based method cannot detect any duplicated regions. However, DCT-based method quickly leads to higher FPR when the SNR level is higher, which indicates it is sensitive to noise adding. FMT-based and DCT-improved have a better performance than the former, however, with the proposed method has the lowest FPR, followed by DCT-improved.

Similar behavior is observed in the case of Gaussian blurring, Fig. 13(c)–(d) give the results, where the tampered images are blurred by a Gaussian filter ($w = 5, \sigma = 0.5, 1, 1.5, 2, 2.5, 3$). The DAR/FPR curves in Fig. 13(c)–(d) show that, the proposed method still has a better performance than other methods. In Fig. 13(c), the DAR curves of the proposed method gains higher performance, with DAR $\geq$ 90%, however, we can also see the DAR curves of DCT-based, DCT-improved, PCA-based and FMT-based drop quickly when the blurring radius increased. FPR curves in Fig. 13(d) also give a satisfactory performance, with the proposed method has the lowest FPR, even though with larger blurring radius ($\sigma = 3$).

From the above analysis, thought it may lead to some false matches, I think based on our approach and human interpretation, we are able to visually confirm a suspicious image.

4. Conclusions

We have presented an automatic and efficient detection algorithm for copy-move forgery. It works without any digital watermarks or signatures information. Compared with previous works, such as [2–6], our approach used less features to represent each block. The experiments show that the proposed algorithm could not only endure the multiple copy-move forgery, but also the blurring or nosing adding and with low computational complexity. Thus, we believe our method can give a little contribution to the area of forensic science.

Acknowledgements

We thank the support from National Science Fund of China (60873117) and Key Program of Natural Science Fund of Tianjin (Grant # 07JCZDJC06600) and Key Program of Natural Science Fund of Tianjin (Grant # 11JCZDJC16000), China.

References

[1] H. Farid, Exposing digital forgeries in scientific images, presented at the Proceedings of the 8th workshop on Multimedia and security, Geneva, Switzerland, 2006.
[2] A. Fridrich, et al., Detection of Copy-move Forgery in Digital Images, 2003.
[3] Y. Huang, et al., Improved DCT-based detection of copy-move forgery in images, Forensic Science International 206 (1–3) (2011) 178–184.
[4] A. Popescu and H. Farid, Exposing digital forgeries by detecting duplicated image regions, Dept. Comput. Sci., Dartmouth College, Tech. Rep. TR2004-515, 2004.

[5] L. Weiqi, et al., Robust Detection of Region-Duplication Forgery in Digital Image, in Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, 2006, pp. 746-749.
[6] S. Bayram, H.T. Sencar, N. Memon, An efficient and robust method for detecting copy-move forgery, in: IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE Press, New York, 2009.
[7] B. Mahdian, S. Saic, Detection of copy-move forgery using a method based on blur moment invariants, Forensic Science International 171 (2007) 180–189.
[8] X. Pan, S. Lyu, Detecting image region duplication using SIFT features, in: IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), 2010, 2010, 1706–1709.
[9] T.-T. Ng, J. Hsu, S.-F. Chang, Columbia image Splicing Detection Evaluation Dataset, http://www.ee.columbia.edu.cn/In/dvmm/downloads/AuthSpliceDataSet/AuthSplicedDataSet.htm/, 2009.
[10] http://r0k.us/graphics/kodak/.