Passive forensics for copy-move image forgery using a method based on DCT and SVD

Jie Zhao a,b,*, Jichang Guo a

a School of Electronic Information Engineering, Tianjin University, Tianjin 300072, China
b School of Computer and Information Engineering, Tianjin Chengjian University, Tianjin 300384, China

A R T I C L E   I N F O

A B S T R A C T

As powerful image editing tools are widely used, the demand for identifying the authenticity of an image is much increased. Copy-move forgery is one of the tampering techniques which are frequently used. Most existing techniques to expose this forgery need to improve the robustness for common post-processing operations and fail to precisely locate the tampering region especially when there are large similar or flat regions in the image. In this paper, a robust method based on DCT and SVD is proposed to detect this specific artifact. Firstly, the suspicious image is divided into fixed-size overlapping blocks and 2D-DCT is applied to each block, then the DCT coefficients are quantized by a quantization matrix to obtain a more robust representation of each block. Secondly, each quantized block is divided non-overlapping sub-blocks and SVD is applied to each sub-block, then features are extracted to reduce the dimension of each block using its largest singular value. Finally, the feature vectors are lexicographically sorted, and duplicated image blocks will be matched by predefined shift frequency threshold. Experiment results demonstrate that our proposed method can effectively detect multiple copy-move forgery and precisely locate the duplicated regions, even when an image was distorted by Gaussian blurring, AWGN, JPEG compression and their mixed operations.

© 2013 Elsevier Ireland Ltd. All rights reserved.

1. Introduction

With the widespread use of powerful digital image editing tools, even people who are not experts in image processing can tamper with an image easily without leaving visible clues. Thus it poses a very serious social problem as to how much of its content can be believed in, whether it is authentic or tampered, especially as a witness in a courtroom, insurance claims and scientific fraud. When the counterfeit images are used for vicious purpose, it may result in inestimable losses. Therefore, developing techniques to verify the authenticity and integrity of digital images become very significant, which is one of the primary goals in image forensics. Over the past few years, the field of digital forensics, which is novel and exciting, has emerged to help restore the reliance to digital images. Digital watermarking [1,2] has been addressed as a means by which an image can be authenticated, however, the drawback of which is that a watermark must be inserted at the time of recording, which would restrict this approach to specially equipped digital cameras. In contrast to this method, passive-blind forensics aims at assessing the authenticity of an image without prior knowledge and in the absence of watermarks or signatures, which works by assuming that even if the tampered images do not reveal any visual artifacts or anomalies, the underlying statistics of these images would be distinct from the original ones. Owing to its incomparable advantage, passive-blind image forensics has become a very hot and promising research aspect in the image authentication field.

Among forgery techniques using typical image processing tools, copy-move forgery is the most common type of image tampering where a region of an image is copied and pasted to another non-intersecting region in the same image to conceal an important element or to emphasize a particular object. Since duplicated regions come from the same image, they have similar properties like texture, noise and color. An example of copy-move forgery is illustrated in Fig. 1. Fig. 1(a) is the original image, while Fig. 1(b) is a fake counterpart where the fountain on the right was copied and placed over the base of the building on the left. In the last decade, many passive detection schemes for copy-move forgery have been proposed, which could be grouped into two categories: block-based methods and keypoint-based methods. Fridrich [3] first analyzed the exhaustive search and then proposed a block matching detection scheme based on discrete cosine transform (DCT) which is one of the landmark methods for copy-move forgery.
forgery detection. Popescu [4] proposed a similar method which used principal component analysis (PCA) instead of DCT. The accuracy of the method is good except for small block sizes and low SNR. Luo [5] extracted color features as well as special intensity ratio to represent a block characteristics vector. A different approach was presented by Kang [6] in which the features were represented by the singular value decomposition (SVD). Bayram [7] applied Fourier-Mellin transform (FMT) to each block and FMT values were finally projected to one dimension to form the feature vector. Mahdian [8] used a method based on blur moment invariants to locate the forgery regions. Li [9] extracted the features of the circular blocks using rotation invariant uniform local binary patterns. Lynch [10] proposed an efficient expanding block algorithm primarily using direct block comparison instead of indirect comparisons based on block features. Almost all the methods mentioned above are block-based which attempt to find an effective and robust representation of each block, moreover, they are expected to be insensitive to common post-processing operations including additive white Gaussian noise (AWGN), Gaussian blurring and JPEG compression.

Unlike block-based methods, keypoint-based methods rely on the identification and selection of high-entropy image regions. In [11–13], some approaches that extracted keypoints by scale-invariant feature transform (SIFT) were proposed to detect the forgery due to their robustness to several geometrical transforms such as rotation and scaling. However, SIFT-based scheme still has a limitation on detection performance since it is only possible to extract the keypoints from peculiar points of the image and not robust to some post-processing operations like blurring and flipping based on our experimental results. Shivakumar [14] proposed another keypoint-based method which used speeded up robust features (SURF). Recently, Chen [15] developed a method by extracting Harris corner points as keypoints and employing step sector statistics to represent the small circle image region around each Harris point. The main drawback of most keypoint-based methods is that copied regions are often only sparsely covered by matched keypoints. Thus they do not provide the exact extent and location of the detected duplicated region, but only displays the matched keypoints. Furthermore, if the copied region exhibits little structure, it may happen that the region is completely missed [16].

Most existing methods are typically evaluated against simple forgeries where human viewers have no trouble to identify the duplicated regions and low resolution images which are a far cry from realistic tampered images with high resolution. In this paper, we develop an effective and robust detection algorithm based on DCT and SVD whose framework is also block-based. A series of experiments conducted on challenging realistic forgery images demonstrate our method can not only effectively detect multiple copy-move forgery and precisely locate the duplicated regions, but also has stronger robustness to common post-processing attacks such as Gaussian bluring, additive white Gaussian noise, JPEG compression and their mixed operations.

The rest of the paper is organized as follows. The review of DCT and SVD is given in Section 2. In Section 3, the proposed forgery detection method is described in detail. The experimental results are given and the corresponding analysis is discussed in Section 4. The conclusion is drawn in Section 5.

2. Review of DCT and SVD

2.1. Discrete cosine transform

Discrete cosine transform (DCT) is a mathematical transformation method which can transform each pixel of an image in the spatial domain into DCT coefficient in the frequent domain. It is significant to numerous applications in science and engineering, from lossy compression of audio (e.g. MP3) and images (e.g. JPEG) to spectral methods for the numerical solution of partial differential equations. It is worthy to note that DCT has some useful properties which are of particular value to image processing, such as energy compaction, de-correlation, symmetry and so on, especially for lossy data compression, because it has a strong “energy compaction” property: most of the signal information tends to be concentrated in a few low-frequency components of the DCT. Two-dimensional DCT of a $M \times N$ matrix is defined as follows:

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos \left( \frac{2m+1} {2M} \right) \cos \left( \frac{2n+1} {2N} \right) \quad 0 \leq p \leq M - 1, \quad 0 \leq q \leq N - 1$$

$$\alpha_p = \left\{ \begin{array}{ll} \frac{1} {\sqrt{M}} & p = 0 \\ \frac{1} {\sqrt{2M}} & 1 \leq p \leq M - 1 \end{array} \right. \quad \alpha_q = \left\{ \begin{array}{ll} \frac{1} {\sqrt{N}} & q = 0 \\ \frac{1} {\sqrt{2N}} & 1 \leq q \leq N - 1 \end{array} \right. \quad (1)$$

The values $B_{pq}$ are called the DCT coefficients of special grayscale value $A_{mn}$.

2.2. Singular value decomposition

Singular value decomposition (SVD) is a matrix factorization and has three properties, namely, stability, scaling property and rotation invariance, which represents algebraic and geometric invariant properties of an image [17,18]. SVD has been used in a
large amount of fields such as signal processing, data compression and pattern analysis. The basic theory of SVD is as follows:

Let \( A \) be an image matrix with \( A \in \mathbb{R}^{M \times N} \) of rank \( r \), its SVD is given by the formula:

\[
A = U \Sigma V^T
\]  

where \( U \) is an \( M \times M \) matrix of the orthonormal eigenvectors of \( AA^T \), \( V^T \) is the transpose of a \( N \times N \) matrix containing the orthonormal eigenvectors of \( A^TA \). \( \Sigma \in \mathbb{R}^{M \times N} \) is a \( M \times N \) diagonal matrix of the singular values which are the square roots of the eigenvalues of \( A^TA \), divided in the form of equation:

\[
\Sigma = \begin{bmatrix}
\Sigma_r & 0 \\
0 & 0
\end{bmatrix}
\]  

where \( \Sigma_r \) is a square diagonal matrix in \( \mathbb{R}^{r \times r} \), \( \Sigma_r = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r) \). \( r \) is the rank of \( A \) that is equal to number of non-negative singular values. The positive diagonal entries in \( \Sigma_r \) are called the singular values of \( A \) and ranked in decreasing order as follows:

\[
\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0.
\]

SVD is not only an important tool for exploratory data analysis, dimensionality reduction and data compression, but also a method for noise reduction. The singular values are unique for a matrix, which form a steady representation of image blocks. Through a large number of experiments, we find that only a few large singular values dominate for most natural images while all the other singular values are quite small. It can be drawn that the relatively small singular values are sensitive to noise while the largest singular value (LSV) contains most energy of each image block and has a good stability even when images suffer from minor distortions. This is the property of SVD of which our algorithm takes advantage.

3. The proposed algorithm

Owing to the nature of copy-move forgery, there must be at least a pair of similar regions in a tampered image, which is the basis of all passive detection algorithms. A natural image, on the contrary, is unlikely to have two large similar regions except for the images that have a large area of smooth region, such as blue sky or green grassland in the image. Hence, the task of passive-blind forensics is to determine whether an image contains large similar regions. Since the shape and size of copied regions are unknown, it is definitely computationally impossible to try to compare every possible pairs of region pixel by pixel. Obviously, it is more effective to divide a forensic image into fixed-sized overlapping blocks and examine whether pairs of blocks are duplicated. The key step is to extract some appropriate and robust features from each block in order to implement an effective detection. Therefore, a good feature can not only represent the whole block, but also has the robustness of common post-processing operations, and what is more, make the detection algorithm have lower computational complexity.

3.1. Algorithm framework

The discussions above draw forth the framework of copy-move forgery detection algorithm, which is also shown in Fig. 2. The whole detection framework is given as follows:

1. Dividing the suspicious image into fixed-size overlapping blocks.
2. Applying 2D-DCT to each block to generate the quantized coefficients by means of quantization.
3. Extracting SVD-based features from each quantized block.
4. Matching similar block pairs.
5. Removing the isolated blocks and output the detection result.

Fig. 2. Framework of the detection algorithm.

3.2. Implementation details

Our detection algorithm includes seven steps, the details of which are given as follows:

Step1: Pre-processing the suspicious image.
Assume that the suspicious image is a gray image \( I \) of the size \( M \times N \). If it is a color image, it is first converted to a grayscale image using the standard formula:

\[
Y = 0.299R + 0.587G + 0.114B
\]

where \( R, G, B \) are three channels of the input color image, \( Y \) is its luminance component.

Step2: Dividing the suspicious image into fixed-size overlapping blocks.
The suspicious image is divided into overlapping blocks with the size of \( b \times b \) pixels by sliding one pixel form upper left corner right down to lower right corner, that is, the adjacent overlapping blocks only have one different row or column. We assume that the size of duplicated region is larger than the size of overlapping block, which is reasonable if account of the fact that a meaningful forged operation must be of a certain area. Each block is denoted as \( B_{ij} \), where \( i \) and \( j \) indicate the starting point of the block’s row and column respectively. The total number of overlapping blocks for a suspicious image of \( M \times N \) pixels is \((M - b + 1) \times (N - b + 1)\).

Step3: Applying 2D-DCT to each block to obtain DCT coefficients with the same size.
Two-dimensional DCT is applied to each block. After that a DCT coefficients matrix with the same size as the block is exploited, which can represent the corresponding block. Here, we assume that the size of each block is \( 8 \times 8 \), thus the size of the DCT coefficients matrix is also \( 8 \times 8 \). Since the method of extracting features of each block in our algorithm is inspired by the thought that a quantization table is applied to each DCT coefficients matrix in the JPEG image encoding, the size of each block is definitely assigned to \( 8 \times 8 \).

Step4: Dividing DCT coefficients matrix by quantization steps stored in the quantization matrix and rounding to integers.
It is the nature of DCT that the energy of transformed domain will be focused on the low frequency coefficients, that is, not all of the elements are equally important and the top-left part of DCT coefficients represents most of the information. We are inspired from quantization mechanism of JPEG images encoding in order to yield a more robust representation of each image block. In our method, the size of each block $B_k$ is set to $8 \times 8$ and the quantized DCT coefficient is computed with

$$D_{ij}^q = \text{round} \left( \frac{D_{ij}}{Q_{ij}} \right) , \quad i, j \in \{0, 1, \ldots, 7\}$$

(5)

where $D_{ij}$ is the un-quantized DCT coefficient, $Q_{ij}$ is the quantization table in which the element $Q_{ij}$ controls the compression ratio, with larger values producing greater compression, and $D_{ij}^q$ is the quantized coefficient.

Step 5: Representing each quantized block by partitioning it into sub-blocks and extracting robust features from each sub-block.

We divide each overlapping quantized block into non-overlapping sub-blocks with the size of $2 \times 2$ pixels. SVD is applied to each sub-block and its largest singular value is recorded. For each quantized block of size $8 \times 8$ pixels, we can obtain $16$ largest singular values of different sub-block which constitute a feature vector to represent the block as shown in Fig. 3. Assuming that $S_{\text{max}}$ is the largest singular value of the corresponding sub-block labeled $p (p \in \{1, \ldots, 16\})$, they can be combined to form a feature vector with the size of $1 \times 16$, denote as:

$$V = [S_{\text{max}1}, S_{\text{max}2}, \ldots, S_{\text{max}16}]$$

(6)

So a $8 \times 8$ block is represented by a $1 \times 16$ feature vector, compared with other block-based detection methods in [3, 4], whose feature dimension is $1 \times 64, 1 \times 32$, the dimension of ours is lower.

Step 6: Matching similar block pairs.

After obtaining the feature vector for each block, a matrix $A$ is created by arranging the feature vectors into the feature matrix $A$. The row of matrix $A$ corresponds to the feature vector extracted of each block and the number of rows in matrix $A$ is $(M - b + 1)(N - b + 1)$ equal to the total number of image blocks. In brief, the feature matrix $A$ is with the size of $(M - b + 1)(N - b + 1) \times 16$.

$$A = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_{(M-b+1)(N-b+1)} \end{bmatrix}$$

(7)

Note that if the two blocks in the suspicious image are similar enough, their corresponding feature vectors in feature matrix $A$ would be similar as well. Thus matching similar block pairs can be conducted by lexicographically sorting the rows of matrix $A$ to make the feature vectors of similar blocks adjacent to each other. Here the lexicographically sorted matrix is denoted as $A$. To avoid some false matches whenever possible, the algorithm needs to pay a close attention to the mutual positions of each matching block pair and outputs a specific block pair only if there are many other matching pairs in the same mutual position, that is, they have the same shift vector. Towards this goal, if two consecutive rows of the sorted matrix $A$ are found, the algorithm calculates the shift vector between them, stores the positions of each block which represented by a feature vector with the size of $1 \times 16$ in a separate list and increments a shift vector counter $C$. Formally, let $(i_1, j_1)$ and $(i_2, j_2)$ be the top-left corner’s coordinate of the two blocks which represented by two consecutive rows of the sorted matrix $A$, thus the shift vector $S$ between them is calculated as:

$$S = (s_1, s_2) = (i_1 - i_2, j_1 - j_2).$$

(8)

Due to the shift vector $-S$ and $S$ correspond to the same shift, the shift vectors are normalized, if necessary, by multiplying by $-1$ so that $S \geq 0$. For each pair of blocks represented by the consecutive rows of the sorted matrix $A$, we increment the normalized shift vector counter $C$ by one:

$$C(S_1, S_2) = C(S_1, S_2) + 1.$$ (9)

The shift vector counter $C$ is initialized to zero before the algorithm starts. At the end of the matching process, the counter $C$ indicates the frequencies with which different normalized shift vectors occur. Then the algorithm finds all normalized shift vectors $S_1, S_2, \ldots, S_K$, whose occurrence exceeds a user-specified threshold $T_{\text{shift}}$:

$$C(S_r) > T_{\text{shift}} \quad \text{for all} \ r = 1, 2, \ldots, K.$$ (10)

It is worth mentioning that the value of the threshold $T_{\text{shift}}$ is related to the size of the smallest region that can be identified by the algorithm. Larger values may cause the algorithm to miss some not-so-closely matching blocks, while too small a value of $T_{\text{shift}}$ may introduce too many false matches.

In addition, since the duplicated regions are assumed to be not overlapping and the divided blocks may be overlapping, an additional user-specified parameter $T_0$ is also applied to make a judgment, that is, shift vectors is counted if and only if it is generated by two similar feature vectors whose Euclidean distance is larger than $T_0$:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} > T_0.$$ (11)

In sum, in order to determine whether the blocks are duplicated or not, we set two thresholds: shift frequency threshold $T_{\text{shift}}$ and distance threshold $T_0$. Only if the test satisfies the Eqs. (10) and (11), we mark a color map for the copied and duplicated blocks to show the forgery detection result.

Step 7: Post-processing and outputting the final detection result.

Morphologically open operation is applied to fill the holes in marked regions and remove the isolated blocks, then output the final detection result.

4. Experiment results and analysis

All the experiments were carried out on the platform with Intel Pentium 2.13 GHz and Matlab R2010b. The tampered images were created by Photoshop CS3 based on the following three datasets. The first dataset contains some test images in common use came from the miscellaneous volume of USC-SIPI database with the sizes of $256 \times 256$ pixels and $512 \times 512$ pixels [19]. The second dataset contains 24 uncompressed PNG true color images of size $768 \times 512$ pixels released by Kodak Corporation for unrestricted research usage [20]. In addition, we collected 50 high resolution images of size $1024 \times 768$ pixels from Google image search [21], which formed the third dataset. In our experiments, without specific specification, all the parameters are set as: $b = 8, Q = 75, T_d = 40, T_{\text{shift}} = 90$ by default. A disk with the radius of 5 is used in the
By using our method, for each image with the four different sizes from the three datasets mentioned above, it takes about 9 s, 36 s, 55 s and 115 s to locate the tampered regions respectively. Nevertheless, if we use C++ or Java programming languages to implement the algorithm, our algorithm will achieve higher efficiency.

4.1. Performance evaluation

For practical applications, the most important aspect of a detection method is the ability to distinguish tampered and original images. However, the power to correctly locate the tampered region is also significant, which gives the strong
evidence to expose digital forgeries. Thus, we evaluate the performance of our algorithm at two levels: at image level, where we focus on whether the fact that an image has been tampered or not can be detected; at pixel level, where we evaluate how accurately can tampered regions be identified.

At image level, we keep a record of some important measures which are the number of correctly detected forged images \( T_0 \), the number of images that have been erroneously detected as forged \( F_T \), and the falsely missed forged images \( F_N \). From these we compute the measures Precision, \( p \) and Recall, \( r \) which are defined as follows [16]:

\[
p = \frac{TP}{TP + FN}, \quad r = \frac{TP}{TP + FN}.
\]  

(12)

Precision denotes the probability that a detected forgery is truly a forgery; while Recall shows the probability that a forged image is detected.

At pixel level, we adopt two quantitative measures to evaluate the performance of the proposed method. Denote \( \psi_S, \psi_T \) as pixels of original region and forgery region in original image respectively, and \( \psi_S, \psi_T \) as pixels of original region and forgery region in detected result image respectively. From these we compute the detection accuracy rate \( DAR \) and the false positive rate \( FPR \). They are defined as follows:

\[
DAR = \frac{\|\psi_S \cap \hat{\psi}_S\| + \|\psi_T \cap \hat{\psi}_T\|}{\|\psi_S\| + \|\psi_T\|},
\]

(13)

\[
FPR = \frac{\|\psi_S - \hat{\psi}_S\| + \|\psi_T - \hat{\psi}_T\|}{\|\psi_S\| + \|\psi_T\|}.
\]

(14)

where \( \| \) means the area of region, \( \cap \) means the intersection of two regions and \( - \) means the difference of two regions. In this sense, \( DAR \) indicates the performance of algorithm correctly locating pixels of copy-move regions in the tampered image, while \( FPR \) reflects the percentage of pixels which are not contained in duplicated region but included by the implemented method. That is, two parameters indicate how precisely our algorithm can locate

**Table 1**

Detection results of the tampered images distorted by Gaussian blurring.

<table>
<thead>
<tr>
<th>w=3, σ=0.5</th>
<th>w=3, σ=1</th>
<th>w=5, σ=0.5</th>
<th>w=5, σ=1</th>
</tr>
</thead>
<tbody>
<tr>
<td>w=32, 32</td>
<td>48 x 48</td>
<td>w=32, 32</td>
<td>48 x 48</td>
</tr>
<tr>
<td>w=5, 32</td>
<td>48 x 48</td>
<td>w=5, 32</td>
<td>48 x 48</td>
</tr>
<tr>
<td>( p )</td>
<td>1.000</td>
<td>0.993</td>
<td>0.985</td>
</tr>
<tr>
<td>( r )</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>( DAR )</td>
<td>0.922</td>
<td>0.986</td>
<td>0.913</td>
</tr>
<tr>
<td>( FPR )</td>
<td>0.033</td>
<td>0.015</td>
<td>0.033</td>
</tr>
</tbody>
</table>
copy-move regions. The more DAR is close to 1 and FPR is close to 0, the more precise the method would be.

4.2. Effectiveness and accuracy test

In the following experiment, we select some color images with the size of 768 × 512 pixels from the second dataset to test the effectiveness of our algorithm. Here, we divide these images into two groups. In the first group, we randomly choose three kinds of blocks with different sizes, which are 32 × 32 pixels, 64 × 64 pixels and 96 × 96 pixels (corresponding to 0.26%, 1.04% and 2.34% of total image area) respectively, to tamper with, while all the duplicated regions are non-regular and meaningful objects in the second group. All the doctored images in this experiment are without any post-processing operation and the corresponding detection results are illustrated in Figs. 4–6. The top row shows the tampered images, with the yellow line indicating the copy-move regions and pasting location, and the bottom gives the detection results. Owing to space constraints, just a part of the experimental results are given here.

Fig. 4 shows that the accuracy rate DAR is generally greater than 0.95 and the false positive rate FPR equals to 0, that is, our algorithm can locate the tampered regions quite precisely. In addition, Fig. 4 also indicates that the detection performance of our algorithm gradually improves with the increase of size of duplicated regions. Fig. 5 illustrates that our algorithm can find the duplicated regions precisely when all the duplicated regions are non-regular and meaning, even though there are large similar or flat regions in the image, such as large areas of sky or water. Due to the homogenous background in the suspicious images, it is challenge to discern the forgery. To the best of our knowledge, a number of previous methods cease to be effective under the circumstances; however, the detection results of our algorithm are satisfactory. Images shown in Fig. 6 are the testing results which demonstrate that our algorithm work well even when the tampered images have multiple duplicated regions, however, literature [3–10] fail to consider such forgery.

4.3. Robustness test

Since forgers usually do their utmost to create an imperceptible tampered image, various kinds of post-processing operations are carried out such as additive Gaussian noise, Gaussian blurring, JPEG compression or mixed operations. In this section, we conduct a series of experiments to test the robustness of the proposed method. Fig. 7 shows such situation, which indicates that our algorithm can locate multiple duplication regions under different post-processing operations with a satisfactory degree, even when the image is processed by mixed operations. However, literature [3,4,6–14] do not give such experiment.

Furthermore, in order to evaluate quantitatively the robustness of our algorithm to different image distortions, we selected randomly 100 original images from the three datasets to generate doctored images by copying a square region at a random location and pasting onto a non-overlapping region. The sizes of square region were 32 × 32 pixels and 48 × 48 pixels respectively, each kind of which included four differently relative location to generate 800 tampered images. These tampered images together with their original version were then distorted by commonly used post-processing operations with different parameters, such as

### Table 2
Detection results of the tampered images distorted by additive white Gaussian noise.

<table>
<thead>
<tr>
<th></th>
<th>SNR = 40 dB</th>
<th></th>
<th>SNR = 35 dB</th>
<th></th>
<th>SNR = 30 dB</th>
<th></th>
<th>SNR = 25 dB</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>32 × 32</td>
<td>48 × 48</td>
<td>32 × 32</td>
<td>48 × 48</td>
<td>32 × 32</td>
<td>48 × 48</td>
<td>32 × 32</td>
<td>48 × 48</td>
</tr>
<tr>
<td>p</td>
<td>0.990</td>
<td>1.000</td>
<td>0.988</td>
<td>0.990</td>
<td>0.968</td>
<td>0.982</td>
<td>0.965</td>
<td>0.978</td>
</tr>
<tr>
<td>r</td>
<td>1.000</td>
<td>1.000</td>
<td>0.990</td>
<td>1.000</td>
<td>0.978</td>
<td>0.985</td>
<td>0.933</td>
<td>0.938</td>
</tr>
<tr>
<td>DAR</td>
<td>0.987</td>
<td>0.995</td>
<td>0.979</td>
<td>0.991</td>
<td>0.932</td>
<td>0.982</td>
<td>0.885</td>
<td>0.893</td>
</tr>
<tr>
<td>FPR</td>
<td>0.003</td>
<td>0.003</td>
<td>0.003</td>
<td>0.066</td>
<td>0.089</td>
<td>0.112</td>
<td>0.169</td>
<td>0.154</td>
</tr>
</tbody>
</table>

### Table 3
Detection results of the tampered images distorted by JPEG compression.

<table>
<thead>
<tr>
<th></th>
<th>Q = 90</th>
<th></th>
<th>Q = 80</th>
<th></th>
<th>Q = 70</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>32 × 32</td>
<td>48 × 48</td>
<td>32 × 32</td>
<td>48 × 48</td>
<td>32 × 32</td>
<td>48 × 48</td>
</tr>
<tr>
<td>p</td>
<td>0.947</td>
<td>0.967</td>
<td>0.903</td>
<td>0.913</td>
<td>0.882</td>
<td>0.899</td>
</tr>
<tr>
<td>r</td>
<td>0.973</td>
<td>0.978</td>
<td>0.923</td>
<td>0.940</td>
<td>0.823</td>
<td>0.874</td>
</tr>
<tr>
<td>DAR</td>
<td>0.961</td>
<td>0.981</td>
<td>0.879</td>
<td>0.944</td>
<td>0.773</td>
<td>0.815</td>
</tr>
<tr>
<td>FPR</td>
<td>0.012</td>
<td>0.003</td>
<td>0.097</td>
<td>0.004</td>
<td>0.139</td>
<td>0.029</td>
</tr>
</tbody>
</table>

![Fig. 8. DAR/FPR curves for DCT, SVD and proposed methods with different Gaussian blurring (w = 5, σ = 0.5, 1, 1.5, 2, 2.5 and 3) when the duplicated region is 64 × 64 pixels.](image-url)
Gaussian blurring, AWGN and JPEG compression. The experimental results were given in Tables 1–3, which evaluated the robustness from image level and pixel level respectively.

The detection results shown in Tables 1–3 indicate that the larger the area of duplicated region is, the better the detection performance would be, no matter which post-operation the image is distorted by. Table 1 shows that the proposed method has a high detection performance when the images are distorted by Gaussian blurring, even when the image has poor quality \((w = 5, \sigma = 1)\) and small region \((32 \times 32 \text{ pixels})\), our method fails to detect only six images in a total of 400 tampered images \((r = 0.985)\). We can draw a conclusion from Table 2 that our method performs well also in the case of processing AWGN distorted images. Results of tampered images distorted by JPEG compression with different quality are shown in Table 3, which indicate that our method has the ability to locate tampered regions in the case of slight compression.

In the last experiment, we compared our method with two relevant approaches: DCT-based [3] and SVD-based [6]. Here we only selected 400 tampered images with the duplicated size of \(48 \times 48 \text{ pixels}\) for convenience. The overall average DAR/FPR performance comparisons over 400 tampered images are shown in Figs. 8–10.

In the case of Gaussian blurring, Fig. 8 indicates that the DAR curve of the proposed method gains better performance than others, with \(\text{DAR} > 85\%\), when the blurring radius increases. The FPR curve also gives a satisfactory result that our method has the lowest FPR, even though with larger blurring radius \(\sigma = 3\). However, keypoint-based methods [11–15] fail to detect such forgery. Similar behavior is observed in the case of noise adding illustrated in Fig. 9, where the tampered images are distorted by white Gaussian noise \((\text{SNR} = 25 \text{ dB}, 30 \text{ dB}, 35 \text{ dB} \text{ and } 40 \text{ dB})\). With the increase of SNR levels, the DAR increases and the FPR decreases for all methods. Observations from the DAR/FPR curves show that, the SVD-based method has the worst performance especially when SNR drops to about 25 dB, while our method achieves higher DAR and lower FPR than other methods. Fig. 10 shows that the comparison result when the tempered images are contaminated by JPEG compression with different quality factor \((Q = 70, 75, 80, 85 \text{ and } 90)\), which illustrates that our method does work well when the images are slightly compressed.

5. Conclusions

We have proposed a robust passive detection method for copy-move forgery which works in the absence of digital watermarks or signatures information. Compare with previous works, such as [3–8], the overall performance of our method is better. The experiment results show that the proposed algorithm could not only effectively detect multiple copy-move forgery and precisely locate the duplicated regions, but also has stronger robustness to Gaussian blurring, AWGN, JPEG compression and their mixed operations. Thus, we believe our study can make a little contribution to the area of multimedia forensics.

Acknowledgements

This work was supported by Higher School Science & Technology Fund Planning Project of Tianjin City (Grant #20120712), China.

References


