A JPEG Image Blind Forensics Method for Copy-Move Forging

Mingwei Cai, Ming Xu, Jian Xu, Haiping Zhang, Ning Zheng
Institute of Computer Application Technology Hangzhou Dianzi University, Hangzhou,China,310018
cmw010411@gmail.com, *mxu@hdu.edu.cn*

Abstract

As a common forging method, copy-move forgery is to hide and cover the certain areas of image by using copies from the same image or other image. This paper introduced a blind forensics approach to detect copy-move forging. Based on the fact of errors caused by copy-move tampering and JPEG compression, an improved SAD (Sum of Absolute Difference) algorithm based on DWT (Discrete Wavelet Transform) is proposed to detect copy-move tampered image. Experimental results show that the proposed method can effectively detect the forged area in copy-move forged image. Furthermore, the proposed method has lower computational complexity and higher accuracy compared with the previous works.

Keywords: *Copy-move Forgery, JPEG, Error Analysis, DWT, SAD*

1. Introduction

Nowadays, with the widespread availability of image processing technology, various kinds of forged images appear in our lives. The digital image forensics techniques are trying to detect the forged images. Current digital image forensic approaches can be divided into active and passive-blind approaches. Active methods mainly refer to the digital watermarking technique. It needs to embed a digital watermark at the source side of images[1]; on the contrary, blind methods can work without any pretreatment. In this work, passive-blind approach is discussed.

Copy-move is a specific tampering method in the area of blind forensics, Alin C Popescu and Hany Farid proposed a method by applying the Principal Component Analysis (PCA) to detect copy-move forgery[2]. Jessica Fridrich and David Soukal utilized DCT (Discrete Cosine Transform) to yield a reduced dimension representation for detecting tampered copy-move images[3]. Then Er. Saiqa Khan and Er. Arun Kulkarni developed a method for detection based on DWT (Discrete Wavelet Transform)[4] and the improved algorithm[5]. In [6], Junfeng He et al. proposed an approach that can detect doctored JPEG images and further locate the tamped parts, but this method is effective only when the compression quality is high. Similarly, [7] aimed at detecting copy-move tampering between different JPEG images with unequal quality factors, but the theoretical analysis is not given, and the complexity of the algorithm is high. Sun et al. took the advantage of discrete wavelet transform and eigenvector decomposition to identify the doctored blocks[14][15].

This paper proposed an improved *SAD* (Sum of Absolute Difference) algorithm which applies DWT (Discrete Wavelet Transform) for detecting tampered image and locating the tampered region. Furthermore, the theoretically analysis of the errors caused by copy-move tampering and JPEG compression were presented.

2. Background

In this section, JPEG compression process is introduced firstly. It is a block-based compression scheme, and the processing unit is each 8×8 block.

As shown in figure 1, the image J is first transformed into DCT domain, and then the frequency coefficient J_{dct} is quantized by the quantization table Q, $round(J_{dct}/Q)$ means rounding operation, which is one of the main reasons of information loss in JPEG compression image. After the entropy encoding/decoding, the compressed file is multiplied by the quantization table Q in order to obtain the de-quantized coefficient J_{dct}', and the pixel value is obtained by IDCT transformation. The error caused by rounding operation is called quantization error. In the following, the errors caused by different tampering will be described.

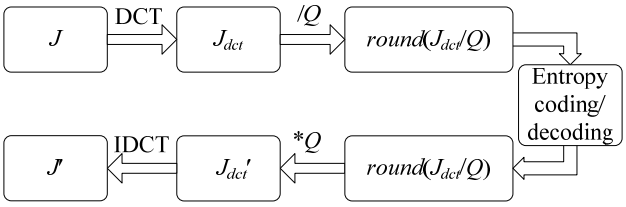
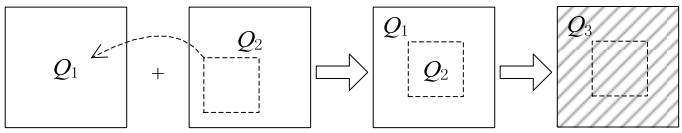


Figure 1. JPEG compression and decompression

Copy-move tampering which is one of the most common types of digital image forging is illustrated in figure 2, Q_1 and Q_2 denote the QF (Quality Factor) of original images respectively.

As shown in figure 2, forgers copy a region from figure 2 (b) and insert it into figure 2 (a), after compositing, figure 2 (c) is obtained, then the composited image is saved by forger in JPEG format (QF=Q_3).



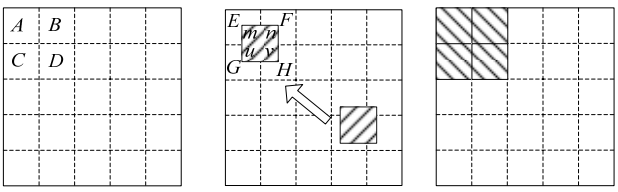
(a) (b) (c) (d)

Figure 2. (a) JPEG image with QF=Q_1 (b) The other JPEG image with QF=Q_2
(c) Synthetic image after copy-move forgery (d) The synthetic image saved in JPEG with QF=Q_3

3. Analyzing the errors caused by tampering

3.1. When quality factor Q_1 equals to Q_2 (Q_1=Q_2)

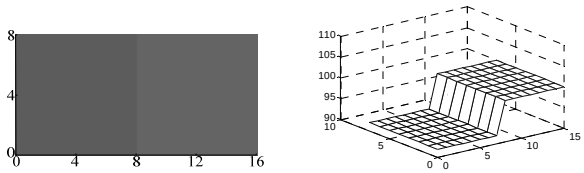
For a simple test under such circumstances, it could be assumed that JEPG image was modified by using sub-parts of itself. It is well known that during JPEG compressing, blocking process will introduce some horizontal or vertical breaks into image. These breaks form a 8×8 grid, called block artifact grid (BAG)[8]. In figure 3, the 8×8 grid is composed by the dotted line. As shown in figure 3 (b), the error is caused by tampering mismatches to the BAG in the neighboring area. Forgers copy the area in the bottom-right corner and then paste it into the top-left corner. In figure 3 (a), we sign the original four 8×8 grids on the top left as *A*, *B*, *C* and *D*. After forging, the remained original part of 8×8 grids is signed as *E*, *F*, *G* and *H*, the tampering part is sliced by the dotted line into four parts: *m*, *n*, *u* and *v*. Figure 3 (c) is obtained by figure 3 (b) after DCT, the dash area on top-left of (c) is the area of block artifacts.



(a) (b) (c)

Figure 3. (a) Original image (b) Tampering in the same image with BAG mismatching
(c) The block artifact caused by resaving in JPEG

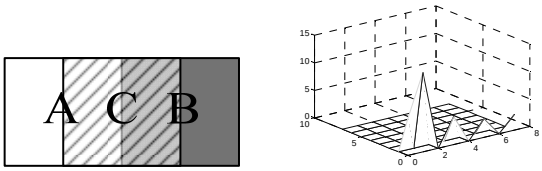
The adjacent 8×8 pixel blocks is signed in figure 4 (a), the 8×16 pixel blocks is chosen from Lena image of JPEG format. Figure 4 (b) is the 3-D coordinate graphs, it can be seen that the pixels have a step between the neighboring 8×8 blocks.



(a) (b)

Figure 4. (a) The neighboring 8×8 blocks (b)The 3-D coordinate of pixels

When Q_1=Q_2, the copy-move tampering will bring the block artifact on the tampered area if the 8×8 block is mismatched. The mismatch tampering is shown in figure 5 (a), A and B is the 8×8 blocks, their pixel values are a and b, C is composed by half of A and B, then the C block is transferred into DCT domain. The 3-D DCT coefficient coordinate is shown in figure 5 (b), The coefficient have zero value on the 2, 4, 6 coordinate point of *x* axis, why does it happen?



(a) (b)

Figure 5. (a) The mismatch tampering (b) The 3-D DCT coefficient coordinate

Formula (1) is the mathematical formula of DCT, $f(i,j)$ is the pixels matrix of C block in figure 5 (a), $F(u,v)$ is the DCT coefficient matrix. From figure 4 (b), $f(i,j)$ could be considered as a step function[13], so formula (1) can be rewritten as formula (2).

$$F(u,v)=\frac{1}{4}C(u)C(v)\left[\sum_{i=0}^{7}\sum_{j=0}^{7}f(i,j)\cos\frac{(2i+1)u\pi}{16}\cos\frac{(2j+1)v\pi}{16}\right] \tag{1}$$

$$C(u),\ C(v)=\begin{cases}\frac{1}{\sqrt{2}}, & \text{if } u=0, v=0\\ 1, & \text{others}\end{cases}$$

$$F(u,v)=\frac{1}{4}C(u)C(v)\left[\sum_{i=0}^{7}\cos\frac{(2i+1)u\pi}{16}\sum_{j=0}^{7}f(i,j)\cos\frac{(2j+1)v\pi}{16}\right] \tag{2}$$

if $u\neq 0$, $\sum_{i=0}^{7}\cos\frac{(2i+1)u\pi}{16}=0$, so $F(u,v)=0$.

if $u=0$, $\sum_{i=0}^{7}\cos\frac{(2i+1)u\pi}{16}=8$, $F(u,v)=2C(u)C(v)\sum_{j=0}^{7}f(i,j)\cos\frac{(2j+1)v\pi}{16}$.

So when $u=0$, the following calculation can be obtained.

$$F(u,v)=2C(u)C(v)\sum_{j=0}^{7}f(i,j)\cos\frac{(2j+1)v\pi}{16}=2C(u)C(v)\left(\sum_{j=0}^{3}a\times\cos\frac{(2j+1)v\pi}{16}+\sum_{j=4}^{7}b\times\cos\frac{(2j+1)v\pi}{16}\right)$$

$$=2(b-a)C(u)C(v)\sum_{j=4}^{7}\cos\frac{(2j+1)v\pi}{16}=2(b-a)C(u)C(v)\left(\cos\frac{9}{16}v\pi+\cos\frac{11}{16}v\pi+\cos\frac{13}{16}v\pi+\cos\frac{15}{16}v\pi\right)$$

From the above formula we can know that if v=2, 4, 6, $F(u,v)$=0, moreover, when v=0, 1, 3, 5, 7 and u=0(the first row of DCT coefficient matrix), $F(u,v)\neq 0$. Due to most DCT coefficient

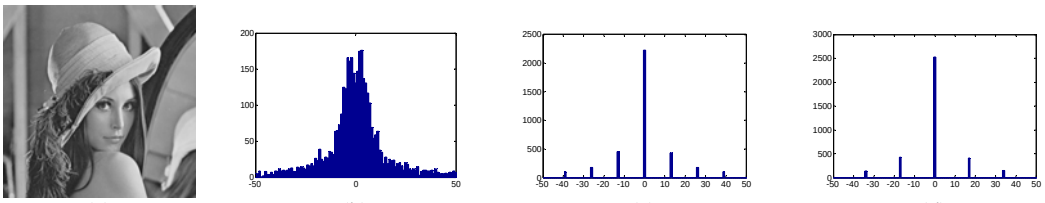
values are zero, the correlation of neighboring blocks is destroyed, so the mismatching copy-move tampering will bring block artifacts. In the tampering, the probability of BAG matching is only about 3.125%, also the BAG matching tampering is poor operational and low application.

The error which is caused by the mismatch of BAG can also appear in different images when Q_1 equals to Q_2. This kind of tampering can be detected by the proposed method. When the copy-move forgery occurs in the different images (quality factor Q_1 not equals to Q_2), the matching probability of BAG is also very low.

3.2. When quality factor Q_1 not equals to Q_2 ($Q_1 \neq Q_2$)

In this case, besides the difference in section 3.1, the compression of different quantization tables brings another kind of differences.

By utilizing quantization steps q_1, q_2 and q_3 to replace quantify factors is much more conveniences for the analysis. In the following, the uncompressed image (Lena.bmp) is utilized to analyze the error caused by tampering. The position (1, 1) of every 8×8 sub-block is chosen after DCT transform (The coordinates of the upper left corner is (0, 0)), the DCT coefficient histogram is shown in figure 6 (b).

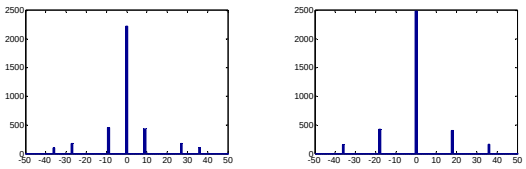


(a) (b) (c) (d)

Figure 6. (a) Lena.bmp (b) DCT coefficient histogram of position (1, 1)
(c) The histogram after quantized by 13 (d) The histogram after quantized by 17

In figure 6 (c) and (d), the histograms exhibit periodic peaks whose periods are 13 and 17 (q_1=13, q_2=17). The distributions of other quantization steps have similar results.

Forged image is resaved in JPEG format, the resaved quality factor is Q_3 (in this section Q_3 is replaced by quantization step q_3). After resaving(q_3=9), the period of histogram is changed (shown in figure 7). Due to this kind of difference, the error caused by compression under different quality factors can be detected.



(a) (b)

Figure 7. (a) DCT coefficient histogram of first quantified by q_1=13, then by q_3=9
(b) DCT coefficient histogram of first quantified by q_1=17, then by q_3=9

The relation among q_1, q_2 and q_3 is described as follows.

$$k_1 = round(n \times q_1 / q_3) \qquad n = 1,2,3... \tag{3}$$

$$k_2 = round(n \times q_2 / q_3) \qquad n = 1,2,3... \tag{4}$$
$$f(k_1) = k_1 \times q_3 \qquad f(k_2) = k_2 \times q_3$$

q_1 and q_2 are the quantization steps of original images, *round*() means rounding operation, n is the multiple of original quantization steps q_1 and q_2, k_1 and k_2 are the distributions after

quantified by q_3, $f(k_1)$ and $f(k_2)$ are the coefficient histogram distributions of JPEG compression by the quantization step q_3.

Suppose the size of uncompressed image I is $m \times n$, the quantization steps are p and q, p is quantization step of the first compression, and q is the second compression.

$$round(I / p) \times p = I' \tag{5}$$

$$round(I' / q) \times q = I'' \tag{6}$$

In (5) and (6), the original image I quantified by p is signed as I', I' quantified by q is defined as I'', the curve graph of $|I''-I'|$ is shown in figure 8, this figure shows the sum of differences between coefficients quantized by the quantization step p=17, followed by a second quantization in the range $q \in [0,100]$, the decimal part of I'/q is λ.

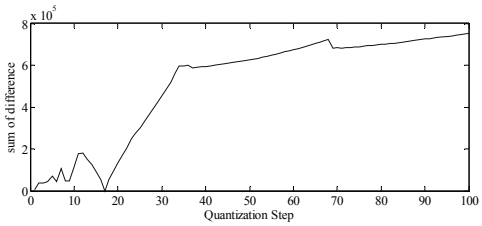


Figure 8. The sum of differences between coefficients quantized by the quantization step p=17, followed by a second quantization in the range $q \in [1,100]$ (horizontal axis) ,this difference reaches a minimum at p=q=17.

Why does the curve of $|I''-I'|$ reach the minimum at q=p=17? It has been discussed that different quantization step brings different errors. This kind of error is brought by operation of rounding. The processing of rounding is described in figure 9 (u=λ, u+v=1), λ is the decimal part of a number, if $\lambda \geqq 0.5$, λ+v (v=1-λ, λ=u), else λ-u (λ=u).

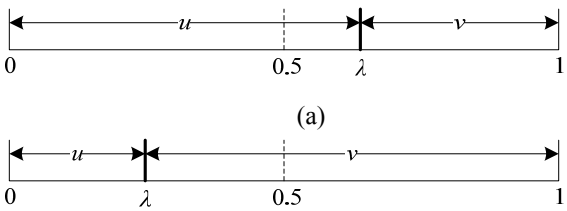


Figure 9. (a) $\lambda \geqq 0.5$, *round(I'/q)=I'/q+v* (b) λ<0.5, *round(I'/q)=I'/q-u*

$$I' = \begin{pmatrix} I_{11} & \cdots & I_{1n} \\ \vdots & \ddots & \vdots \\ I_{m1} & \cdots & I_{mn} \end{pmatrix}, round(I'/q) = \begin{pmatrix} I_{11}/q \pm \min(u_1, v_1) & \cdots & I_{1n}/q \pm \min(u_1, v_n) \\ \vdots & \ddots & \vdots \\ I_{m1}/q \pm \min(u_m, v_1) & \cdots & I_{mn}/q \pm \min(u_m, v_n) \end{pmatrix} \tag{7}$$

In (7), u_i and v_j (i=1,2,3…m, j=1,2,3…n) are the specific performance of u and v in figure 9.

$$I'' - I' = \begin{pmatrix} \pm\min(u_1, v_1) \times q & \cdots & \pm\min(u_1, v_n) \times q \\ \vdots & \ddots & \vdots \\ \pm\min(u_m, v_1) \times q & \cdots & \pm\min(u_m, v_n) \times q \end{pmatrix} \tag{8}$$

$$\sum\sum |I'' - I'| = \sum_{i=1}^{m}\sum_{j=1}^{n} |I''(i,j) - I'(i,j)| = \sum_{i=1}^{m}\sum_{j=1}^{n} |\min(u_i, v_j) \times q| \tag{9}$$

$|I''-I'|$ gets the minimum only when $\min(u_i, v_j)\times q$ obtains the least value, from formula (3) and (4) that when $p=q$, $|I''-I'|$ gets the minimum.

From the demonstration of copy-move tampering in figure 2, the quantization step of original image is q_1=13, q_2=17, the synthesized image is saved by the quantization step q_2=9. Through exhausting JPEG compression (quantization step ranging from 1 to 30 in a step of 1), it can be observed that the difference between curves (solid and dot, shown in figure 10) reaches the peak when the exhausting quantization step near 17.

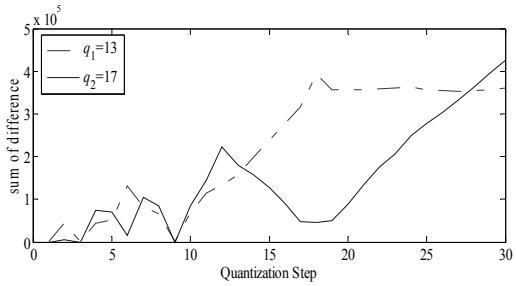


Figure 10. The sum of differences between coefficients quantized by the quantization step q_1=13, q_2=17, followed by a second quantization q_3=9, then exhausting quantizing in the range of [1,30]

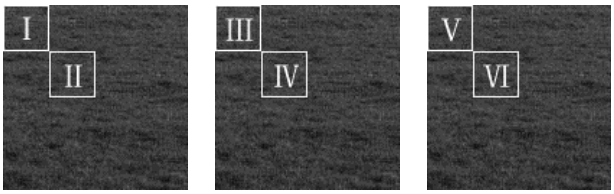
The cause of errors is theoretically analyzed in the above. The following will analyze the errors by experimental results.

NSR (noise signal ratio) which shows the difference between original image and tampered image is described in formula (10).

$$NSR(I_1, I_2) = 10 \times \frac{\sum_{i=a}^{a+m}\sum_{j=b}^{b+n}\left|I_1(i,j) - I_2(i,j)\right|}{\sum_{i=1}^{m}\sum_{j=1}^{n} I_1(i,j)} \tag{10}$$

I_1 and I_2 are the chosen images, $I_1(i,j)$ and $I_2(i,j)$ are the pixels of two different images, a and b are the starting coordinates of the *NSR* compared region, $m\times n$ is the selected contrast area size. Through computing the Ⅰ~Ⅵ pane in figure 11 (a), (b) and (c) by formulas (11), the distinct of *NSR* is obtained.

$$\eta_1 = \frac{NSR(\text{Ⅰ}, \text{Ⅴ})}{NSR(\text{Ⅰ}, \text{Ⅲ})}, \quad \eta_2 = \frac{NSR(\text{Ⅱ}, \text{Ⅵ})}{NSR(\text{Ⅱ}, \text{Ⅳ})} \tag{11}$$



(a) (b) (c)

Figure 11. (a) The tampered image which is uncompressed, the Ⅰ pane is the forged area (from another JPEG image, QF=8), the Ⅱ is the unmodified area (QF=7)
(b) The tampered image is compressed by QF=10. (c) The image (b) is resaved in QF=7

This paper chooses 20 common Matlab images to compute *NSR* error, the tampered area size from (16, 16) to (64, 64) is selected, *m* and *n* are the length and width of the tampered region. Matlab is used in this experiment for tampering and resaving, the quality factor is from 1 to 100. According to figure 2, the experiment can be divided into two groups, group A: Q_1=70, Q_2=80, Q_3 =95; group B: Q_1=70, Q_2=70, Q_3 =95.

In order to take full account of tampering, group A and group B are further divided into three sub cases: A_0 indicates the *NSR* error between non-tampered regions, A_1 indicates the BAG mismatching tampering area, A_2 indicates the BAG matching tampering area, the definition of B_0, B_1, B_2 are the same as A_0, A_1, A_2.

The data of table 1 can be obtained by calculating the six kinds of *NSR* error. In table 1, the η_1/η_2 value of A_1 and A_2 is much larger than A_0, this due to case A_1 brings quantization errors and block artifacts in the tampered area, case A_2 brings quantization errors in the tampered area. In group B, due to Q_1=Q_2, there is no quantization error caused by the different quantization factors, the η_1/η_2 value of B_1 is significantly larger than the value of B_0 and B_2, this is because in case B_1 tampering with BAG mismatching causing the block artifacts. Quantization errors and block artifacts as traces of tampering can be detected by exhausting compression to the tampered image.

Table 1. *NSR* error comparison

Error value	A_0	A_1	A_2	B_0	B_1	B_2
η_1	0.253	3.422	10.460	0.018	2.705	0.424
η_2	0.072	0.118	0.117	0.036	0.150	0.437
η_1/η_2	3.514	29.000	89.402	0.500	18.033	0.970

4. The proposed algorithm

The proposed method first resaves the tampered image *J* with a different QF by exhausting approach, then Discrete Wavelet Transform (DWT) is applied to the tampered image and resaved image. Finally an improved *SAD* algorithm is used between the DC coefficients A and A′, after this the *SAD* image is obtained, and then by image processing, we can get the last detection result. The detecting step is shown in figure 12.

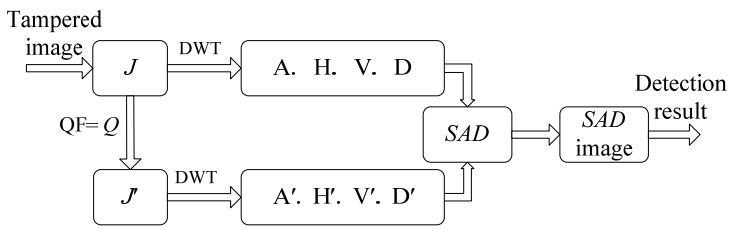


Figure 12. The procedure of detection tampered images.A, H, V, D and A′, H′, V′, D′ mean the four components of *J* and *J′* after DWT.

Formula (12) is the improved *SAD* algorithm, *c*=1, 2, 3 denotes three color channels of RGB (for grayscale image, *c*=1), $A(i+m,j+n,c)$ and $A'(i+m,j+n,c)$ are the DC coefficients in the *c* channel, $p(m,n)$ is the sampling function.

$$SAD(i,j)=\sum_{m=0}^{b-1}\sum_{n=0}^{b-1}\sum_{c=1}^{3}\left|A'(i+m,j+n,c)-A(i+m,j+n,c)\right|\times p(m,n) \tag{12}$$

$$p(m,n)=\begin{cases}1, & m,n \text{ are even number}\\ 0, & \text{others}\end{cases}$$

It is hard for detector to judge the detection results, so a detection result assessment method is proposed, the steps are as follows.

i) First of all, the obtained *SAD* image should undergo binary processing, when $SAD(i,j)<[\max(SAD)+\min(SAD)]/4$, let this point's *SAD* value equals to zero, otherwise equals to 255, $SAD(i,j)$ means the *SAD* value in the (i,j) point of *SAD* image, $\max(SAD)$ indicates the maximum value in *SAD* image, $\min(SAD)$ indicates the minimum value in *SAD* image.

ii) If all the values of generated binary *SAD* image equals to zero, considering the detection failed, otherwise, calculating entropy of the binary *SAD* image. If the entropy is larger than t_1, considering the detection failed, If the entropy is less than t_2, and larger than t_3, it should repeat the operation in step i , modify the binary determine threshold to $[\max(SAD)+\min(SAD)]/6$, then calculating entropy of the binary *SAD* image, if the entropy less than t_2, or larger than t_1, determining the detection failed, if the entropy is larger than t_2 and less than t_1, determining the detection succeed. Entropy is the results of the assessment, threshold t_1, t_2, t_3 respectively setting as 0.45, 0.15, 0.04.

In order to verify the assessment method, we selected 100 non-tampered images with size of 640×480 and 100 the same size images with the tampered region of 140×140. PhotoShop is selected as the experimental tool, which employs a 12-point quality scale. Non-tamper original image is first saved as JPEG image of the quantization factor Q_1, and then double compressed by quantization factor Q_2; the same processing is done in the tampering images. In this experiment, Q_1 is selected as 7, Q_2 is selected as 11. The final results of the assessment are shown in table 2, the value in table is the entropy of binary *SAD* image. It can be seen,that this assessment method can effectively distinguish the tampered and non-tampered images.

Table 2. Comparison of assessment results

Image type	maximum	minimum	average	accuracy rate
Non-tampered	0	0	0	100%
Tampered	0.565	0.009	0.276	97%

5. Experiment and discussion

To evaluate the efficacy of the proposed algorithm, 1000 uncompressed TIFF images were obtained from the UCID dataset[10] (denoted as UCID), Columbia Uncompressed Image Splicing Detection Evaluation Dataset [11] (denoted as Columbia) and our own dataset (denoted as Our Dataset).

In our algorithm, the value of block size *b* can be arbitrarily chosen. But if *b* is too small, the detected tampered region will not be clearly enough to differentiate it from the rest region. When *b* is too large, the edge of detection result will be blurry. In this paper, block size *b* is chosen as 8. The PS (Photoshop) was used for forging and saving images.

5.1. Detecting copy-move tampering from image itself

In this section, the forged image is generated by copying the tampered area from the original JPEG image (QF=Q_1), and pasting it into the image itself, then the entire image was saved at the QF=Q_2. In detecting steps, the tampered image is resaved in JPEG image as QF= Q_1.

Table 3 lists the detection accuracy. From it, all datasets can be effectively detected by proposed method, especially when Q_1 and Q_2 have a large difference

Table 3. Detection accuracy.

Dataset type	Q_1=7,Q_2=10	Q_1=8, Q_2=10	Q_1=9, Q_2=10
UCID	88.4%	88.2%	76.8%
Columbia	94.4%	87.5%	65.2%
Our Dataset	90.8%	85.3%	62.5%

(a) (b) (c)

Figure 13. (a) Original image (b) Doctored image (c) Last detection result

Figure 13 shows an original and doctored image. The tampered image is generated by inserting a piece of image of original image (QF=8) to the image itself, then saving the composite image as a JPEG image with QF=12. In detecting steps, the tampered image is resaved in JPEG as QF=8. As known from figure 13, the proposed algorithm can easily detect copy-move tampering in the image itself.

5.2. Detection of copy-move tampering from the different images

Differ from section 5.1, in this section, the tampered image is generated by copying a central portion from one image (QF=Q_2) and inserted into another image (QF=Q_1), then the composite image was saved at the quality of Q_3. In detecting steps, the tampered image is resaved in JPEG as QF=Q_1.

Table 4 lists the detection accuracy comparison of Hany Farid's method[12]. From the table, in different dataset the proposed work has an effectively detection result despite Q_1, Q_2, and Q_3 have low difference.

Table 4. Comparison of detection accuracy

Dataset type	Detection Method	Q_1=6, Q_2=8, Q_3=10	Q_1=7, Q_2=9, Q_3=10	Q_1=8, Q_2=9, Q_3=10
UCID	Proposed	91.2%	87.2%	86.8%
	H. Farid's	78.5%	74.8%	77.5%
Columbia	Proposed	90.4%	88.9%	89.2%
	H. Farid's	81.2%	78.4%	76.0%
Our Dataset	Proposed	94.7%	91.7%	90.4%
	H. Farid's	87.3%	79.2%	77.6%

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

Figure 14. (a) The original image (b) Another original image
(c) The doctored image (d) Last detection result

Figure 14 shows a doctored image obtained by inserting a cat of figure 14 (b) with QF=10 into figure 14 (a) with QF=8, then the composite images is saved as a JPEG image with QF=11, shown in figure 14 (c). In detecting steps, the tampered image is resaved in JPEG image as QF=8. After detection, figure 14 (d) is obtained by the process of corroding and dilating from *SAD* image. It can be seen that the proposed algorithm can easily detect copy-move tampering in the images with different quality factors.

The proposed method also has a strong robustness, even tampered image is suffered from JPEG compression, adding noise or cutting processing, the method still has a good detection result. Figure 15 (a) is the original image with QF=8; figure (b) is the tampered image saves in JPEG with QF=11, then upload it on the internet; figure (c) is obtained by cutting the tampered image on the webpage by screenshots tool; figure (d) is the detection result; it can be seen that even after image cutting, the proposed method has a good detection results. Figure 16 (a) is the tampered image after QF=8 compression attack, (b) is the detection result; figure (c) is the tampered image after adding Gaussian noise whose variance is 0.02, (d) is the detection result.

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

Figure 15. (a) The original image (b) The tampered image
(c) The tampered image after cutting (d) Last detection result

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

Figure 16. (a) The tampered image after JPEG compression (b) Last detection result
(c) The tampered image after adding noise (d) Last detection result

Because of DWT and sampling function, the time complexity of proposed algorithm is lower than Hany Farid's method in [12]. In formula (12), the low frequency component of DWT can reduce the computation to 1/4 than Hany Farid's, and the sampling function can also lessen calculation to 1/4, so the computational complexity of proposed method is 1/16 than Hany Farid's at least. Meanwhile, the proposed method saves more time than *SAD* method. Table 5 shows the comparison results about time complexity in different size of images.

Table 5. Comparison of detection time (ms)

Algorithm type	3240×4320	2592×1944	1152×768
Hany Farid's	707955	150477	14611
SAD method	102208	30124	3257
The proposed	24922	5755	723

6. Conclusion and future work

In this paper, an improved *SAD* (Sum of Absolute Difference) algorithm based on DWT (Discrete Wavelet Transform) is proposed to detect copy-move tampered image. This method has lower computational complexity and higher accuracy compared with the algorithm proposed in [12]. Besides, the errors caused by BAG mismatch and the compression of different quantization tables are analyzed. The algorithm also works for images in which pasted regions have undergone any transformation such as rotation, scaling, etc., and it can detect relatively small regions that have been altered. The disadvantage of this approach is that it is inaccurate when the tampered image is saved at lower quality than original image. Future work is to discover other methods to overcome this problem.

7. Acknowledgment

This paper is supported by NSFC (No. 61070212, No.61003195), Natural Science Foundation of Zhejiang Province, China (No. Y1090114) and the State Key Program of Major Science and Technology (Priority Topics) of Zhejiang Province, China (No 2010C11050).

8. References

[1] Mahdian B, Saic S, "Blind methods for detecting image fakery", IEEE International Carnahan Conferenceon on Security Technology, 2008, 280-286.
[2] Popescu A C, Farid H, "Exposing digital forgeries by detecting duplicated image regions", Dept. Comput. Sci., Dartmouth College, Tech. Rep., 2004, TR2004-515.
[3] Fridrich J, Soukal D and Lukas J, "Detection of copy-move forgery in digital images", presented at the Digital Forensic Research Workshop, 2003, 6-8.
[4] Er.S K, Er.A K, "An efficient method for detection of copy-move forgery using discrete wavelet transform", International Journal on Computer Science and Engineering, 2010, 2(5): 1801-1806.
[5] Khan S, Kulkarni A, "Reduced time complexity for detection of copy-move forgery using discrete wavelet transform", International Journal on Computer applications, 2010, 6(7): 31-36.
[6] He J F, Lin Z C, Wang L F and Tang X O, "Detecting doctored JPEG images via DCT coefficient analysis", in Proc. Eur. Conf. Computer Vision, Graz, Austria, 2006, 423-435.
[7] Liu Z L, Li X H and Zhao Y Q, "Passive detection of copy-move tampering for digital image forensics", International Conference on Intelligent Computation Technology and Automation, 2011, 649-652.
[8] Li W H, Yuan Y and Yu N H, "Passive detection of doctored JPEG image via block artifact grid extraction", Signal Processing, 2009, 1821-1829.
[9] Luo W Q, Qu Z H, Huang J W and Qiu G P, "A novel method for detecting cropped and recompressed image block", IEEE International Conference on Acoustics, Speech and Signal Processing, 2007, II-217-II-220.
[10] Schaefer G, Stich M, "UCID-An Uncompressed Color Image Database", Technical Report, School of Computing and Mathematics, Nottingham Trent University, 2003.
[11] Ng T T, Chang Shi-F, "A data set of authentic and spliced image blocks", Technical Report, DVMM, Columbia University, 2004. Dataset: http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/photographers.htm.
[12] Farid H, "Exposing digital forgeries from JPEG ghosts", IEEE Transactions on Information Forensics and Security, 4(1): 154-160, 2009.
[13] Zeng B, "Reduction of blocking effect in DCT-coded images using zero-masking techniques", Signal Processing, 1999, 79(2): 205-211.
[14] Michael Z, Sun X M, "DWT-PCA (EVD) Based Copy-move Image Forgery Detection", JDCTA: International Journal of Digital Content Technology and its Applications, Vol. 5, No. 1, pp. 251-258, 2011.
[15] Michael Z, Sun X M, "Fast and Robust Image Cloning Detection using Block Characteristics of DWT Coefficients", JDCTA: International Journal of Digital Content Technology and its Applications, Vol. 5, No. 7, pp. 359-367, 2011.