



## Detecting copy-move forgery under affine transforms for image forensics <sup>☆</sup>



Leida Li <sup>a,\*</sup>, Shushang Li <sup>a</sup>, Hancheng Zhu <sup>a</sup>, Xiaoyue Wu <sup>b</sup>

<sup>a</sup> School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China

<sup>b</sup> No. 29 Institute, China Electronics Technology Group Corporation, Chengdu 610036, China

### ARTICLE INFO

#### Article history:

Available online 15 December 2013

### ABSTRACT

In copy-move forgery, the copied region may be rotated and/or scaled to fit the scene better. Most of the existing methods fail when the region is subject to affine transforms. This paper presents a method for detecting this kind of image tampering based on circular pattern matching. The image is first filtered and divided into circular blocks. A rotation and scaling invariant feature is then extracted from each block using Polar Harmonic Transform (PHT). The feature vectors are then lexicographically sorted, and the forged regions are detected by finding the similar block pairs after proper post-processing. Experimental results demonstrate the efficiency of the method.

© 2013 Elsevier Ltd. All rights reserved.

### 1. Introduction

With the prevalence of powerful image processing softwares, it is easy to tamper an image without leaving visible traces. However, the misuse of these tampered images misleads our understanding of the true image. Image forensics is an emerging technique that can determine the authenticity of an image directly from the data itself [1–3].

Copy-move is a simple and effective way of image tampering, which copies part of an image and pastes it to another part of the same image. The idea of copy-move detection is to divide the image into blocks and find similar block pairs. Recently, various methods have been proposed to solve the problem. Fridrich et al. first proposed to detect this kind of forgery in [4], where the quantized Discrete Cosine Transform (DCT) coefficients were employed as the feature of the block. Popescu and Farid adopted the Principle Component Analysis (PCA) to describe the contents of the block [5]. Forgery detection was achieved by comparing the PCA features. Luo et al. computed seven characteristic features based on statistical analysis of the pixels in each block [6]. The method is effective when the copied region is subject to blur, noise and JPEG compression. In [7], blur moment invariants were used to extract the block features. This method aims to resist the blur post-processing. Huang et al. improved the DCT based method by introducing a truncating procedure to reduce the feature dimension [8]. Cao et al. also proposed an improved DCT based method, where four features were further constructed to reduce the dimension of the feature vector [9]. Although these methods can detect the copy-move forgery in most cases, they fail when the copied region is rotated, scaled or flipped before being pasted.

In order to detect the copy-move forgery under geometric transforms, several methods have been reported recently. Amerini et al. addressed a scale-invariant feature transform (SIFT) based method [10]. It can detect the forged regions when the copied region is subject to rotation and scaling. A possible deficiency of the method is that the forged regions are only marked by the matched feature points. In [11], Hu moments were employed to extract the block features. The method is robust to moderate region rotation and common signal processing operations. Ryu et al. employed the Zernike moment

<sup>☆</sup> Reviews processed and recommended for publication to Editor-in-Chief by Deputy Editor Dr. Gregorio Martinez.

\* Corresponding author. Tel.: +86 15005203739.

E-mail address: [reader1104@hotmail.com](mailto:reader1104@hotmail.com) (L. Li).

in block feature extraction [12]. They built a database with 12 images, and the method achieved an average detection precision rate of 83.59% for region rotation. In [13], log-polar mapping was utilized to deal with the copy-move forgery affected by reflection, rotation and scaling.

In copy-move forgery, affine transforms are usually employed to fit the scene better. Meantime, they bring trouble to the forgery detection. While some of the current methods can handle region rotation and scaling, more general affine transforms can be adopted to produce the forgery images, such as rotation plus scaling, shearing and perspective transform. To this end, this paper presents a circular pattern matching method for detecting the copy-move forgery affected by the affine transforms. The image is first divided into overlapping circular blocks, and Polar Harmonic Transform (PHT) is employed to extract the invariant features. Then the features are lexicographically sorted, and block matching is achieved by comparing the Euclidean distances of the feature vectors. A post-processing filter is proposed to remove the false matches and morphological operation is employed to obtain the final detection result.

The rest of this paper is organized as follows. Section 2 introduces the basic principles of PHT, with emphasis on the Polar Sine Transform (PST). The proposed method is described in Section 3, followed by the experimental results in Section 4. Finally, Section 5 concludes.

### 2. Feature extraction in circular domain

Polar Harmonic Transform is a two dimensional orthogonal moment method [14]. PHT is defined on the unit disk, and it can be obtained by projecting the image onto the orthogonal kernel:

$$M_{nl} = \Omega \int_0^{2\pi} \int_0^1 [H_{nl}(r, \theta)]^* f(r, \theta) r dr d\theta, \tag{1}$$

where  $H_{nl}(r, \theta)$  is the kernel function,  $n$  is the order,  $l$  is the repetition,  $\Omega$  is a constant. The orthogonal kernel consists of a radial component and an angular component:

$$H_{nl}(r, \theta) = R_n(r) e^{il\theta}, \tag{2}$$

where  $R_n(r)$  is the radial part and  $e^{il\theta}$  is the angular part. The kernels are orthogonal and they satisfy the following orthonormal condition:

$$\int_0^{2\pi} \int_0^1 H_{nl}(r, \theta) [H_{pq}(r, \theta)]^* r dr d\theta = \delta_{np} \delta_{lq}, \tag{3}$$

where  $\delta$  is the Kronecker delta defined by

$$\delta_{kk'} = \begin{cases} 1, & k = k' \\ 0, & k \neq k'. \end{cases}$$

According to the kernels used, PHT can be defined in three different forms, namely Polar Complex Exponential Transform (PCET), Polar Cosine Transform (PCT) and Polar Sine Transform (PST). Their kernels are defined as follows:

$$H_{nl}(r, \theta) = \begin{cases} e^{i2\pi nr^2} e^{il\theta}, & \text{for PCET} \\ \cos(\pi nr^2) e^{il\theta}, & \text{for PCT} \\ \sin(\pi nr^2) e^{il\theta}, & \text{for PST.} \end{cases} \tag{4}$$

It has been shown that PST owns the best invariance to geometric distortions [15]. For an image in polar coordinates,  $f(r, \theta)$ , the PST of order  $n$  with repetition  $l$ ,  $n = 1, 2, \dots, \infty, |l| = 0, 1, \dots, \infty$ , is defined as:

$$M_{nl} = \frac{2}{\pi} \int_0^{2\pi} \int_0^1 [\sin(\pi nr^2) e^{il\theta}]^* f(r, \theta) r dr d\theta. \tag{5}$$

Eq. (5) is defined for continuous images. For an  $N \times N$  digital image  $f(x, y)$ , the PST can be approximately calculated as:

$$M_{nl} = \frac{8}{\pi N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} [H_{nl}(x, y)]^* f(x, y). \tag{6}$$

PST has a desirable property that its magnitude is invariant to rotation. Suppose that  $f^R(r, \theta)$  is an image obtained by rotating  $f(r, \theta)$  clockwise by  $\phi$  degree, the PSTs computed on the two images are then related by:

$$M_{nl}^R = M_{nl} \cdot e^{-il\phi}. \tag{7}$$

It is known from Eq. (7) that  $|M_{nl}^R| = |M_{nl}|$ , namely image rotation has no effect on the magnitudes of the PSTs. The magnitudes can also be made to be invariant to image scaling by making the computation area cover the same content. In implementation, this can be done by transforming the image into the uniform domain  $[-1 \ 1] \times [-1 \ 1]$ .

**Table 1**  
Magnitudes of the PSTs for image Lena and the distorted versions.

Distortion	$M_{11}$	$M_{21}$	$M_{23}$	$M_{31}$	$M_{33}$	$M_{35}$
Rotation 0	13.508	7.313	3.747	4.315	4.678	2.130
Rotation 10	13.502	7.312	3.747	4.311	4.674	2.129
Rotation 30	13.505	7.313	3.741	4.314	4.674	2.128
Rotation 60	13.505	7.313	3.750	4.316	4.679	2.133
Rotation 90	13.508	7.313	3.747	4.315	4.678	2.130
Scaling 0.8	13.219	7.380	3.623	4.194	4.783	2.164
Scaling 0.9	13.453	7.360	3.440	3.910	4.796	2.199
Scaling 1.1	13.507	7.318	3.744	4.477	4.484	2.362
Scaling 1.2	13.623	7.251	3.654	4.288	4.531	2.394

Table 1 lists the magnitudes of PSTs for the standard image Lena and some distorted versions. It is known that for the listed distortions, the magnitudes change very little.

### 3. Copy-move forgery detection

The proposed method consists of four steps. The first step is to apply low-pass filtering to the image and divide it into overlapping circular blocks. Then the PSTs are computed for each block, and the feature vectors are lexicographically sorted. The third step is to compare the feature vectors and search for similar block pairs. The last step is to reduce the false matches using the proposed post-processing filter and morphological operations, producing the final detection map.

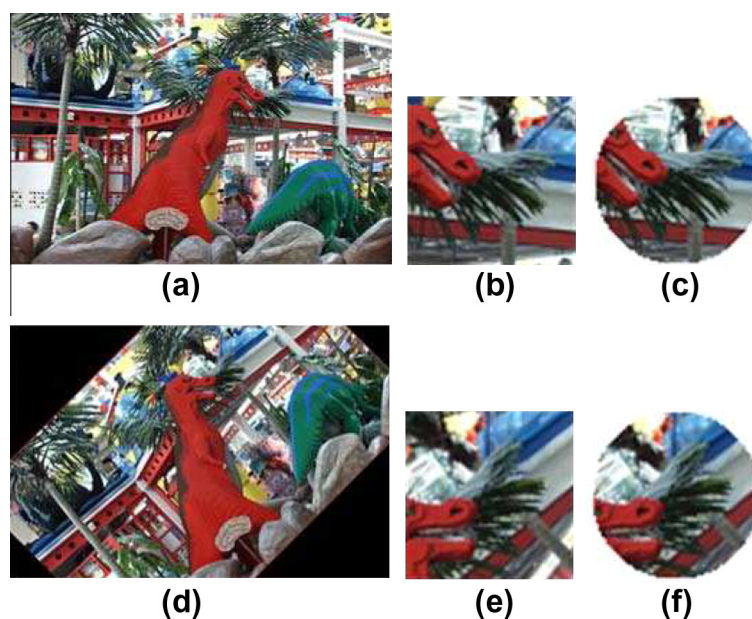
#### 3.1. Pre-processing of the image

The proposed method operates in the luminance domain. Therefore, the color image is first converted into gray scale by

$$I = 0.299R + 0.587G + 0.114B. \quad (8)$$

where  $R$ ,  $G$ ,  $B$  denote the red, green and blue components of the image.

It is well-known that high frequency components are not stable when the image is subject to signal processing operations. Therefore, low frequency features are more useful in the subsequent feature matching. In view of this, we apply a low-pass filter to reduce high frequency disturbances. In implementation, the Gaussian low pass filter is adopted. The standard deviation of the filter is 3.0 and the filter size is  $3 \times 3$ .



**Fig. 1.** Difference between square block and circular block. (a) Original image, (b) square block in (a), (c) circular block in (a), (d) image rotated by 45°, (e) square block in (d), (f) circular block in (d).

In order to detect the forged regions, the filtered image is divided into small circular blocks. Unlike most of the existing methods that use square blocks, circular blocks are employed here. Fig. 1 illustrates the difference between square block and circular block. Fig. 1(a) and (d) shows the original image and the image rotated by 45°. A square block and a circular block are extracted from the two images respectively.

It is known from the figure that the contents keep constant for the circular blocks. However, the contents of the square blocks are different, especially at the four corners. If square blocks are employed, the features extracted from the blocks will be different and the probability that these two blocks match correctly is low. By contrast, the contents of the circular blocks are constant even the image is rotated. Therefore, if a rotation invariant feature extraction method is employed to produce the block features, they can be elegantly matched. To this end, the image is divided into overlapping circular blocks before further processing. The adjacent blocks have only one different row or column. For a  $W \times H$  image, this produces  $(W - 2r + 1) \times (H - 2r + 1)$  circular blocks, where  $r$  is the radius of the block.

### 3.2. Feature extraction

Polar Sine Transform is employed to extract features from the circular blocks. Typically, a feature vector is extracted from a block, so  $(W - 2r + 1) \times (H - 2r + 1)$  features can be obtained. Then they are arranged in a matrix  $S$ . It is straightforward to know that similar blocks should have similar feature vectors. However, if the matrix  $S$  is directly used to perform the block matching, the computation cost will be extremely high. The reason is that the blocks are obtained in a raster scaling order, so the feature vectors of similar blocks are far away in  $S$ . In order to improve the computational efficiency, the matrix is lexicographically sorted. In this way, similar features will be rearranged in the neighboring rows, which facilitates fast block matching.

### 3.3. Block matching

Block matching is to find the similar block pairs by estimating the Euclidean distances of the feature vectors. Let  $S_i$  and  $S_j$  denote the  $i$ th row and  $j$ th row of  $S$ , then the Euclidean distance is computed as follows:

$$D(i, j) = \sqrt{\sum_{k=1}^L [S_i(k) - S_j(k)]^2}, \quad (9)$$

where  $L$  is the dimension of the feature vector. Block matching starts from the first row of the matrix  $S$ . For  $S_i$ , the distances with the following  $R_{lim}$  features are computed, and the one with the minimum distance is obtained:

$$D(i, i + K) = \min\{D(i, i + 1), D(i, i + 2), \dots, D(i, i + R_{lim})\} \quad (10)$$

where  $(i + K)$  is the row that has the minimum distance with  $S_i$ . In order to determine if these two features are correctly matched, a similarity threshold is adopted, which is denoted by  $T_s$ . If  $D(i, i + K)$  is smaller than  $T_s$ , the  $i$ th block and the  $(i + K)$ th block are successfully matched. Their indices are then saved in a set  $\Omega$ . Otherwise, no match is found for  $S_i$ , and it is removed from  $S$ . This process is repeated for all features in  $S$ . Finally, all matched block pairs are recorded in  $\Omega$ .

### 3.4. Post-processing

Since the matched block pairs have been recorded in  $\Omega$ , it is easy to obtain the detection result by marking the block pairs. Most of the existing methods generate the detection map in a block manner, namely the blocks are entirely marked to generate the detection map. This may produce coarse boundary of the detected regions. In order to obtain finer boundaries, we propose to mark only the innermost five pixels for each block. Fig. 2(b) shows an example of the initial detection result using the proposed marking method.

Typically, the initial detection map contains some false results, which usually exist in a pattern somewhat like white noises. In this paper, they are removed using a post-processing filter and morphological operations. The proposed post-processing filter operates as follows. An  $8 \times 8$  sliding window moves in a raster scanning order. Each time, the window

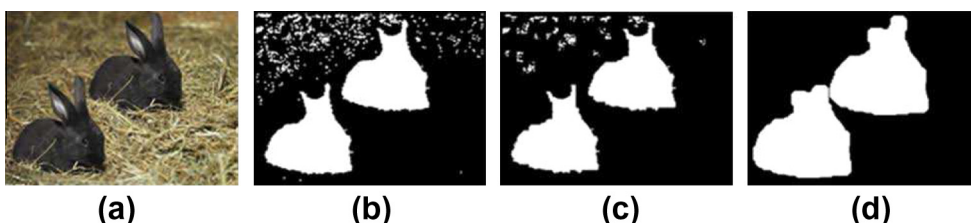


Fig. 2. Post-processing of the detection result. (a) Forged image, (b) initial detection result, (c) result after filtering, (d) result after morphological operation.

moves forward by 8 pixels so that the window covers a new part of the image. At each position, the *white* pixels in the window are counted. If there are less than 20 *white* pixels in the window, the whole window area is marked black. This operation can remove the small isolated false matches. Fig. 2(c) shows the detection map after post-processing filtering. Finally, morphological processing is employed to obtain the final detection map. Specifically, morphological erosion is first applied, followed by morphological dilation. Fig. 2(d) shows an example of the final detection map.

#### 4. Simulation results and discussion

The images used in the experiments are collected from the internet. In implementation, the order of the PST is 3, which can produce 9 dimensional block features. The diameter of the circular block is 16. The similarity threshold is set to  $T_s = 2.1$ , which is determined by experiments. In block matching, the searching range is  $R_{lim} = 30$ . Fig. 3 shows two of the test images, the forged images and the detection results. It is known from the figure that when no attacks are applied, the detection results are very accurate. The detection map shows the forged regions clearly.

##### 4.1. Performance on affine transforms

The proposed method is characterized by the ability to handle affine transforms. In the subsequent experiments, the copied regions are rotated, scaled or flipped before they are pasted to the destination regions. The simulation results are shown in Figs. 4–9. For comparison, the simulation results of Liu's method [11] and Ryu's method [12] are also included.

Region scaling is first tested. Figs. 4 and 5 show the detection results for the rabbit and dinosaur images, where the objects are scaled with factors 0.7, 0.9 and 1.1, respectively.

It is known from the figures that the proposed method can handle region scaling efficiently, and the detection maps clearly show the tampered regions. Although the contents of the fixed-size circular blocks will be different during region scaling, the extracted block features change very little, because PST is a statistical feature extraction method. By comparison, Liu's method fail to detect the forged regions. While Ryu's method can also handle region scaling, the proposed method achieves the highest accuracy, which can be seen from the detection maps.

Next, we conduct an experiment on region rotation. Fig. 6 shows the simulation results when the object is rotated by  $15^\circ$ ,  $30^\circ$  and  $90^\circ$ , respectively.

It can be easily seen that the Hu moment based method fail to detect the forged regions when the rabbit image is rotated by  $15^\circ$  or  $30^\circ$ . By comparison, the Zernike based method and the proposed method can both detect the forgery successfully. Furthermore, the proposed method can locate the tampered region with higher accuracy. It is also interesting to find that when the region is rotated by  $90^\circ$ , three methods performs competitively. It is not hard to understand, because when the region is rotated by  $90^\circ$  or integral multiple of 90, there is no interpolation error. As a result, feature extraction and feature matching are both accurate.

We also test the performance of the proposed method under combined attacks. Fig. 7 shows the tampered images and the detection maps, where rotation plus scaling is applied to the test image. The results show that Hu moment based method is inefficient in dealing with this kind of distortion. The Zernike moment based method can detect the forged regions to some extent. The proposed method performs best.

Fig. 8 shows the simulation results on region flipping. Region flipping is easy to detect because there is no interpolation error in flipping. Three methods can all detect the forged regions successfully and accurately.

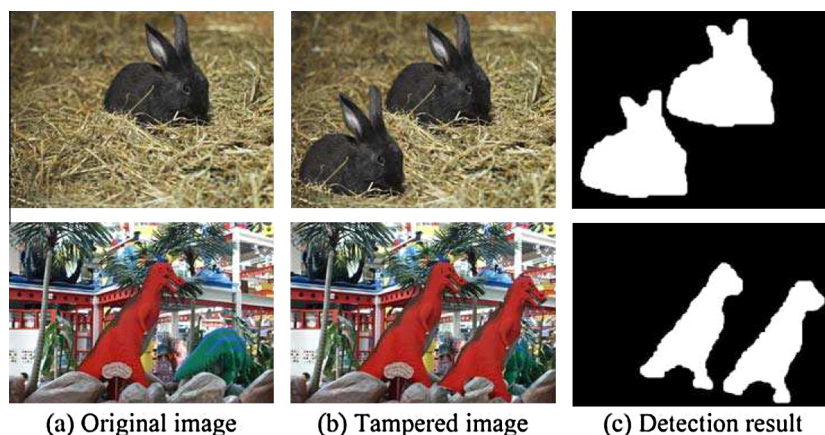


Fig. 3. Detection results on the tampered images without attacks.

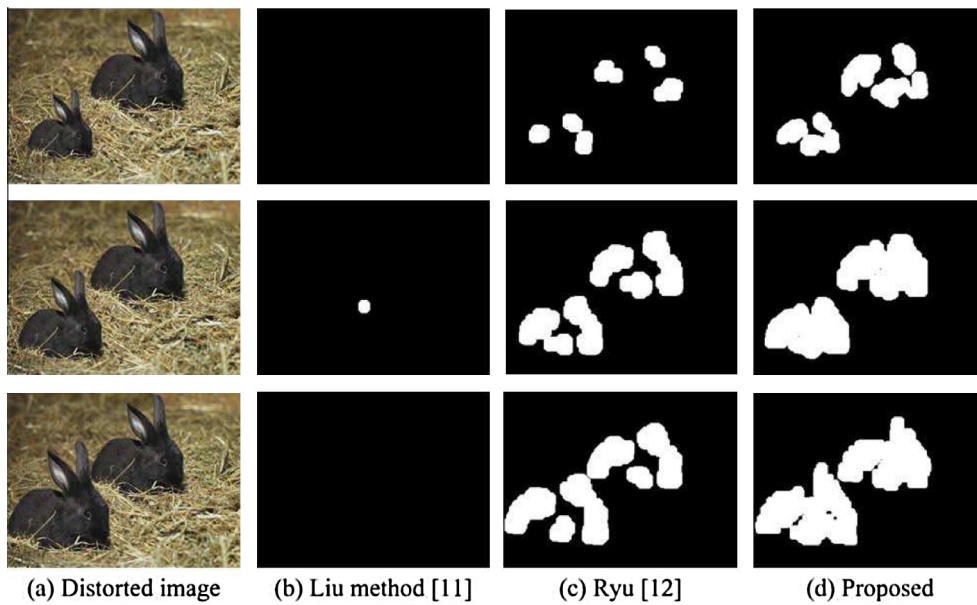


Fig. 4. Detection results of region scaling on the rabbit image. Top: scaling 0.7 $\times$ ; Middle: scaling 0.9 $\times$ , Bottom: scaling 1.1 $\times$ .

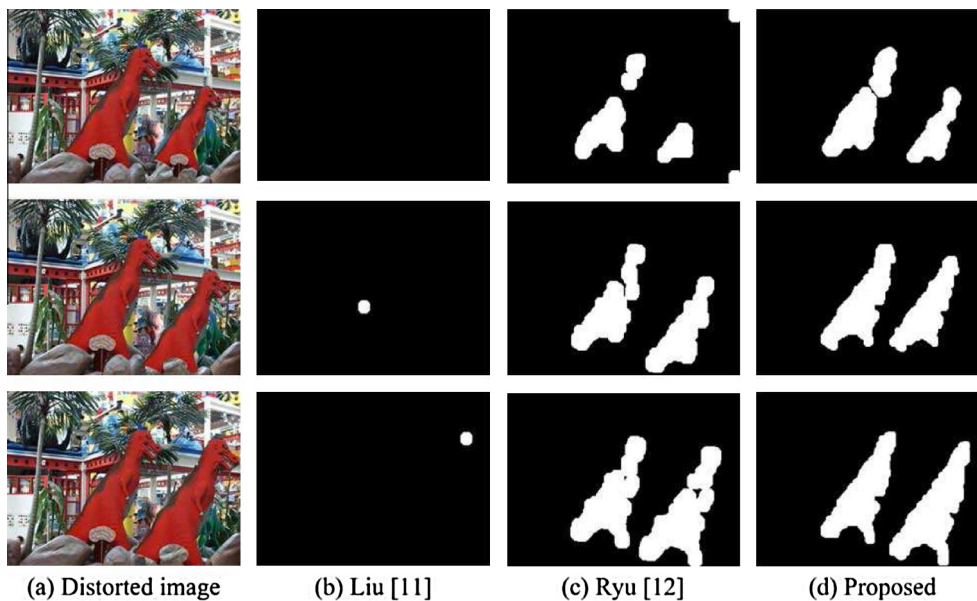


Fig. 5. Detection results of region scaling on the dinosaur image. Top: scaling 0.7 $\times$ ; Middle: scaling 0.9 $\times$ , Bottom: scaling 1.1 $\times$ .

The last experiment in this section is to test the performance of the method to the general affine transforms, including shearing and perspective transform. This kind of region processing has rarely been addressed before. For shearing, the region is sheared both horizontally and vertically by 20°. For perspective transform, the region is distorted by two sets of different parameters. The simulation results are shown in Fig. 9.

Affine transforms pose great challenge to the detection method, because the feature extraction method should be highly resistant to the attacks. It is observed from the figure that Zernike based method and the proposed method can both detect this kind of attacks. By comparison, Ryu's method produces more false detections, and the proposed method performs better.

From the above experiments, it can be briefly summarized that the proposed method is efficient in handling the copy-move forgery when the general affine transforms are applied. Compared with Liu's and Ryu's methods, the proposed method can achieve higher detection accuracy.

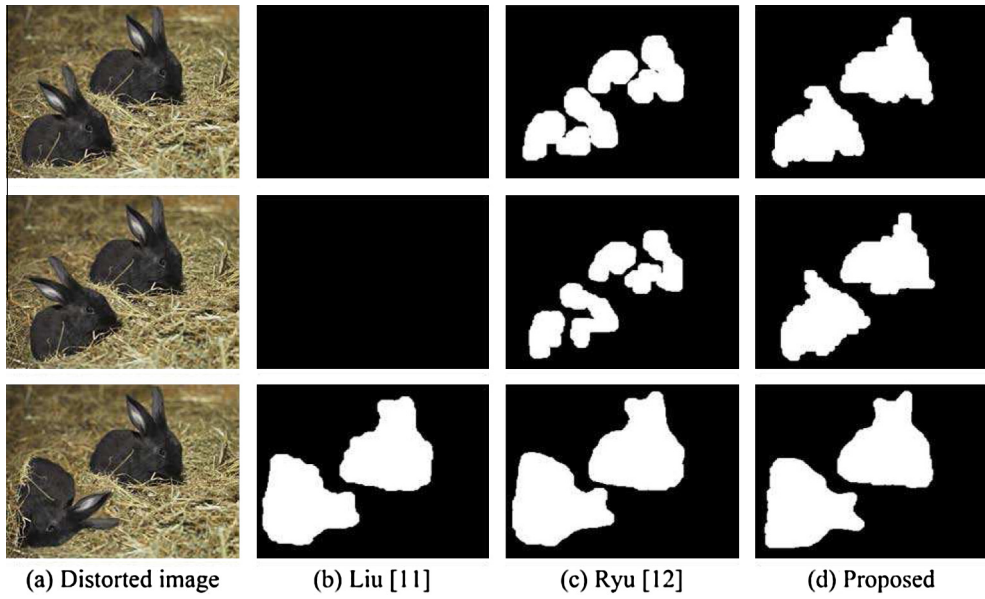


Fig. 6. Detection results on region rotation. Top: rotation 15, Middle: rotation 30; Bottom: rotation 90.

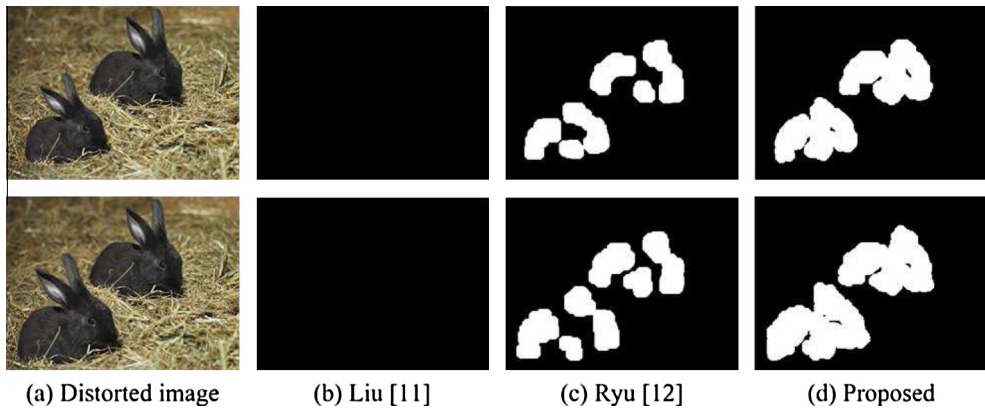


Fig. 7. Detection results on combined geometric attacks. Top: rotation 15 + scaling 0.9×; Bottom: rotation 15 + scaling 1.1×.

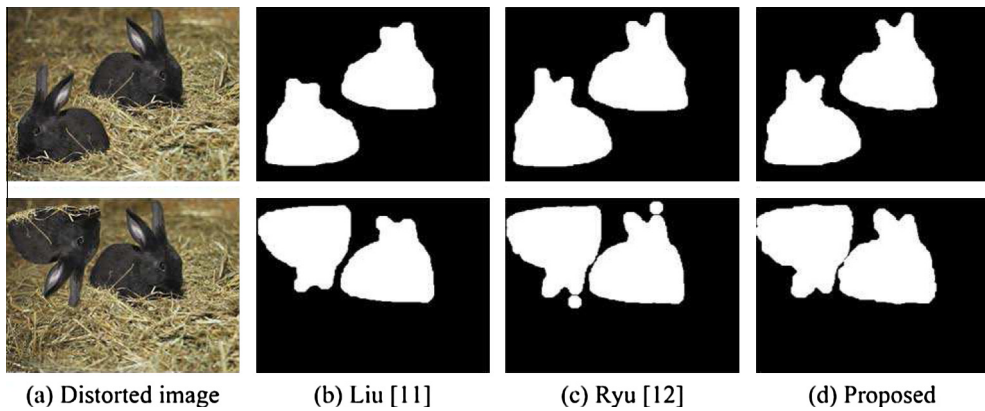
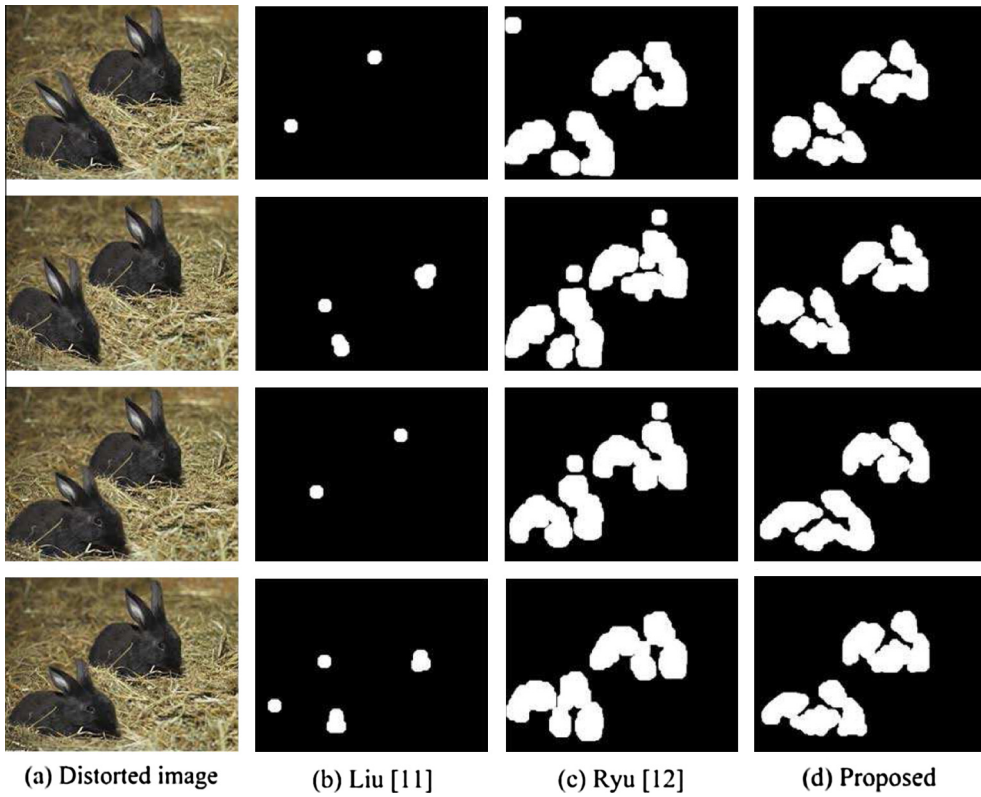
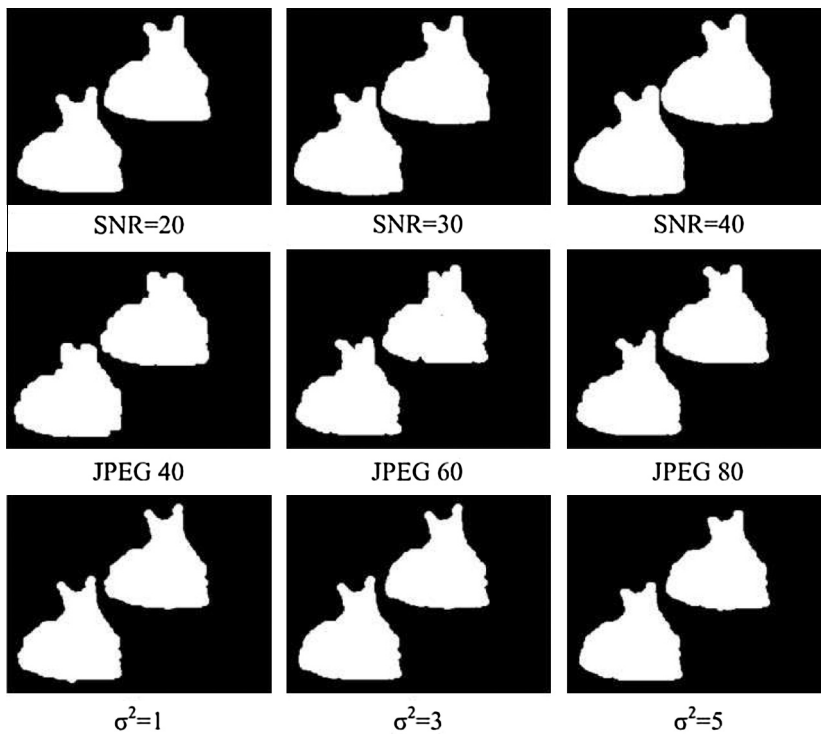


Fig. 8. Detection results on region flipping. Top: horizontal flipping; Bottom: Vertical flipping.



**Fig. 9.** Detection results on affine attacks. From top to bottom: Horizontally shearing 20, Vertically shearing 20, Perspective transform case 1, Perspective transform case 2.



**Fig. 10.** Detection results on signal processing attacks. Top: added white Gaussian noise (AWGN), Middle: JPEG compression, Bottom: Gaussian blur.

#### 4.2. Performance on signal processing attacks

The ability to resist signal processing attacks is fundamental to the copy-move detection methods. Fig. 10 shows the simulation results of the proposed method on added white Gaussian noise (AWGN), JPEG compression and Gaussian blur. For AWGN, the signal to noise ratios (SNRs) of the images are 20, 30 and 40 respectively. For JPEG compression, three different quality factors are employed, namely 40, 60 and 80. For Gaussian blur, the size of the window is  $3 \times 3$ , and the standard deviations are 1, 3 and 5.

It is seen from Fig. 10 that the proposed method can detect the forgery accurately for the signal processing attacks. For AWGN, the SNRs do not have much effect on the detection results. For JPEG compression, the detection results tend to be worse for low quality factors. However, we find that the proposed method can successfully detect the forgery even the quality factor falls as low as 20. Gaussian blur also has very little effect on the detection results. By experiments, we find that the proposed method is very robust to the signal processing operations.

#### 4.3. Quantitative analysis

In this section, the correct detection ratio  $F_c$  and the false detection ratio  $F_f$  are employed to evaluate the overall performance of the proposed method:

$$F_c = \frac{|C_1 \cap C_2| + |M_1 \cap M_2|}{|C_1| + |M_1|}, \quad (11)$$

$$F_f = \frac{|C_1 \cup C_2| + |M_1 \cup M_2|}{|C_1| + |M_1|} - F_c, \quad (12)$$

where  $C_1$  is the copy region,  $M_1$  is the tampered region, while  $C_2$  and  $M_2$  are the detected copy region and the detected tampered region respectively.

The experiments are conducted on two image databases. Database I, is the UCID – Uncompressed Color Image Database [16], which contains more than 1300 images. We randomly choose 100 images to perform the experiment. Database II, is built with 100 images from the internet. In implementation, an  $80 \times 80$  block is copied and pasted to another region of the same image. Then the forged images are subject to the signal processing attacks. The correct detection ratios and the false detection ratios are listed in Table 2. Note that each value is an average of 100 images.

It is known from the table that the correct detection ratios are very high. Even when the forged images are compressed with quality factor 40, the correct detection ratios are also higher than 0.92. Meantime, most of the false detection ratios are lower than 0.1, indicating that the results are promising.

In the next experiment, a block is copied and pasted to another region of the same image. Then the image is processed by JPEG compression, Gaussian blur and added white Gaussian noise. The correct detection ratios and the false detection ratios are then computed for each image from the two databases. Then the average values are computed to generate the curves between correct detection ratios/false detection ratios and signal processing operations. Figs. 11–13 show the simulation results. Note that three different block sizes are tested, including  $60 \times 60$ ,  $80 \times 80$  and  $100 \times 100$ .

It is observed from the figures that the correct detection ratio improves when the JPEG quality factors increases. Meantime, the false detection ratio drops with increasing quality factors. When the quality factor is higher than 40, the correct detection ratios are higher than 0.85, while the false detection ratios are lower than 0.1. In practice, the JPEG quality factors are usually higher than 50, so the proposed method is effective in handling JPEG compression. Gaussian blur and AWGN have limited effect on the performance of the proposed method, which can be seen from Figs. 12 and 13. It is also observed that bigger regions are easier to detect.

**Table 2**

Correct detection ratio and false detection ratio on two databases.

Attack	Database I		Database II	
	$F_c$	$F_f$	$F_c$	$F_f$
AWGN (SNR = 15 dB)	0.9753	0.0931	0.9965	0.0812
AWGN (SNR = 20 dB)	0.9989	0.1048	0.9989	0.0958
AWGN (SNR = 30 dB)	0.9989	0.1048	0.9989	0.0959
AWGN (SNR = 40 dB)	0.9989	0.0957	0.9989	0.0958
JPEG compression 80	0.9943	0.0605	0.9925	0.0719
JPEG compression 60	0.9847	0.0731	0.9901	0.0843
JPEG compression 40	0.9286	0.1378	0.9748	0.1189
Gaussian blurring ( $\sigma^2 = 1$ )	0.9959	0.0397	0.9959	0.0401
Gaussian blurring ( $\sigma^2 = 3$ )	0.9959	0.0394	0.9959	0.0398
Gaussian blurring ( $\sigma^2 = 5$ )	0.9959	0.0406	0.9959	0.0396

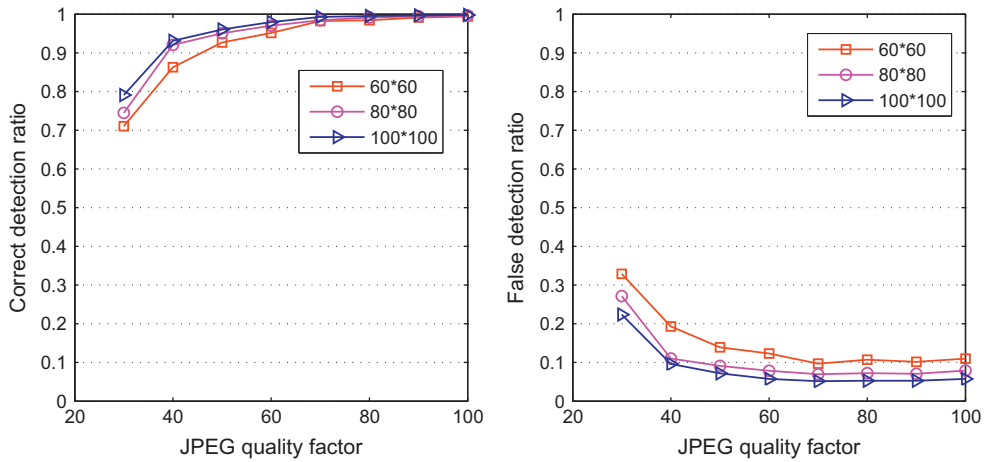


Fig. 11. Performance on JPEG compression.

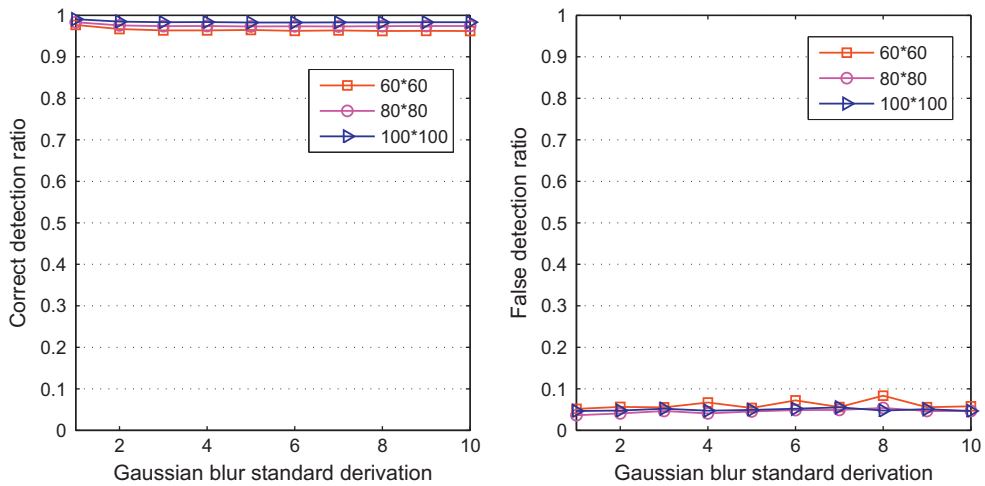


Fig. 12. Performance on Gaussian blur.

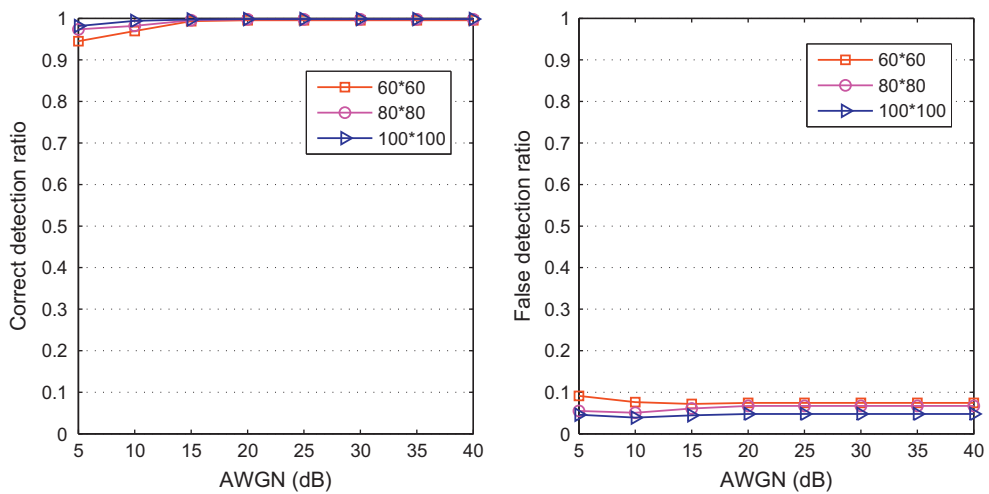


Fig. 13. Performance on added white Gaussian noise (AWGN).

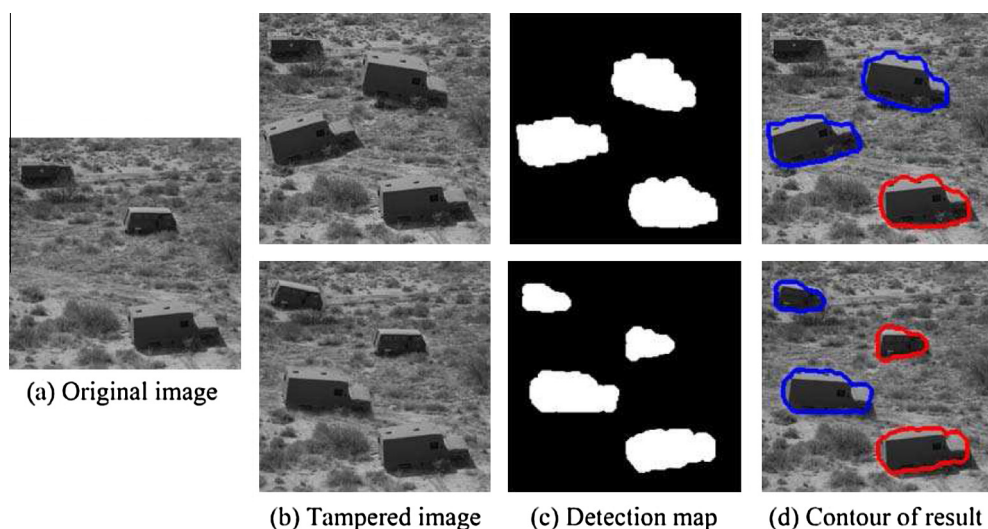


Fig. 14. An example to detect multiple copy-move forgery.

#### 4.4. Extension to multiple copy-move detection

The proposed method can also detect the multiple copy-move forgery. In an image, the object may be copied and pasted to several different regions of the image. Alternatively, several regions may be copied and pasted in the same image. The proposed method can also detect these kinds of forgeries. Fig. 14 shows an example of multiple copy-move detection. In the first case, the truck at the bottom right corner is copied. Then it is rotated clockwise by  $15^\circ$  and anti-clockwise by  $10^\circ$  respectively, producing two distorted regions. Finally, they are pasted to two different areas of the image. In the second case, the truck at the bottom right corner is copied and pasted to the left side after rotation of  $10^\circ$ . The small truck at the upper right corner is copied, rotated by  $10^\circ$ , and pasted to cover the small truck at the upper left corner. The experimental results are shown in Fig. 14(c) and (d).

## 5. Conclusion

Copy-move is a common method to create forgery images. In practice, the copied region may be first transformed before being pasted. The key to detect this kind of image tampering is to extract invariant features from the local image blocks. The proposed method achieves this goal by extracting invariant moment features from the local circular blocks. The proposed method can resist both the traditional signal processing operations and affine transform of the copied regions. The performance of the proposed method is validated by experiments.

In practice, malicious attackers tend to adopt more sophisticated techniques to tamper an image. For example, the copied region may be first transformed by geometric distortions. Then additive noise and blur may be used to hide the traces of tampering. Finally, the image can be saved in JPEG format. In such cases, detection of the copy move forgery is more challenging and thus better feature extraction methods are desired.

## Acknowledgments

This work is supported in part by National Natural Science Foundation of China (61379143, 51204175, 51204176 and U1261105) and the Fundamental Research Funds for the Central Universities (2012QNA59).

## References

- [1] Farid H. Image forgery detection. *IEEE Signal Proc Mag* 2009;26:16–25.
- [2] Mahdian B, Saic S. A bibliography on blind methods for identifying image forgery. *Signal Process – Image* 2010;25:389–99.
- [3] Redi JA, Taktak W, Dugelay JL. *Digital image forensics: a booklet for beginners*. *Multimed Tools Appl* 2011;51:133–62.
- [4] Fridrich J, Soukal D, Lukas J. Detection of copy-move forgery in digital images. In: *Proceedings of the digital forensic research workshop*. Cleveland, OH, USA; 2003. p. 55–61.
- [5] Popescu AC, Farid H. Exposing digital forgeries by detecting duplicated image regions. Technical Report TR2004-515, Dartmouth College; 2004.
- [6] Luo WQ, Huang JW, Qiu GP. Robust detection of region-duplication forgery in digital image. In: *Proceedings of 18th international conference on pattern recognition*. Hong Kong, China; 2006. p. 746–9.
- [7] Mahdian B, Saic S. Detection of copy-move forgery using a method based on blur moment invariants. *Forensic Sci Int* 2007;171:180–9.
- [8] Huang YP, Lu W, Sun W, Long DY. Improved DCT-based detection of copy-move forgery in images. *Forensic Sci Int* 2011;206:178–84.
- [9] Cao YJ, Gao TG, Fan L, Yang QT. A robust detection algorithm for copy-move forgery in digital images. *Forensic Sci Int* 2012;214:33–43.

- [10] Amerini I, Ballan L, Caldelli R, Bimbo AD, Serra G. A SIFT-based forensic method for copy-move attack detection and transformation recovery. *IEEE Trans Inf Foren Sect* 2011;6:1099–110.
- [11] Liu GJ, Wang JW, Lian SG, Wang ZQ. A passive image authentication scheme for detecting region-duplication forgery with rotation. *J Netw Comput Appl* 2011;34:1557–65.
- [12] Ryu SJ, Lee MJ, Lee HK. Detection of copy-rotate-move forgery using Zernike moments. In: *Proceedings of the 12th information hiding conference*. Alberta, Canada; 2010. p. 51–65.
- [13] Bravo-Solorio S, Nandi AK. Automated detection and localisation of duplicated regions affected by reflection, rotation and scaling in image forensics. *Signal Process* 2011;91:1759–70.
- [14] Yap PT, Jiang XD, Kot AC. Two-dimensional Polar Harmonic Transforms for invariant image representation. *IEEE Trans Patt Anal Mach Intell* 2010;32:1259–70.
- [15] Li LD, Li SS, Wang GH, Abraham A. An evaluation on circularly orthogonal moments for image representation. In: *Proceedings of international conference on information science and technology*. Nanjing, China; 2011. p. 394–7.
- [16] Schaefer G, Stich M. UCID – an Uncompressed Colour Image Database. In: *Proceedings of SPIE storage and retrieval methods and applications for multimedia*. San Jose, USA; 2004. p. 472–80.

**Leida Li** received the Ph.D. degree from Xidian University in 2009. Currently, he is an associate professor with School of Information and Electrical Engineering, China University of Mining and Technology. His research interests include image forensics and media quality assessment.

**Shushang Li** received the M.S. degree from China University of Mining and Technology in 2013. He is now a Ph.D. candidate with School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. His research interests include image forensics and compressed sensing.

**Hancheng Zhu** is now a master student with School of Information and Electrical Engineering, China University of Mining and Technology. His research interest is image quality assessment.

**Xiaoyue Wu** received the Ph.D. degree from Xidian University in 2010. He is now an engineer at No.29 Institute, China Electronics Technology Group Corporation. His research interests include digital image processing and multimedia systems.