J. Vis. Commun. Image R. 31 (2015) 320–334

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

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

Copy-move image forgery detection based on Gabor magnitude ☆

Jen-Chun Lee

Department of Electrical Engineering, Chinese Naval Academy, Kaohsiung 813, Taiwan

ARTICLE INFO

Article history:
Received 10 December 2014
Accepted 9 July 2015
Available online 15 July 2015

Keywords:
Digital image forensics
Image splicing
Copy-move forgery
Image retouching
Histogram of orientation Gabor magnitude
Gabor filter
Duplicate region detection
Lexicographical order

ABSTRACT

With advancement of media editing software, even people who are not image processing experts can easily alter digital images. Various methods of digital image forgery exist, such as image splicing, copy-move forgery, and image retouching. The most common method of tampering with a digital image is copy-move forgery, in which a part of an image is duplicated and used to substitute another part of the same image at a different location. In this paper, we present an efficient and robust method to detect such artifacts. First, the tampered image is segmented into overlapping fixed-size blocks, and the Gabor filter is applied to each block. Thus, the image of Gabor magnitude represents each block. Secondly, statistical features are extracted from the histogram of orientated Gabor magnitude (HOGM) of overlapping blocks, and reduced features are generated for similarity measurement. Finally, feature vectors are sorted lexicographically, and duplicated image blocks are identified by finding similarity block pairs after suitable post-processing. To enhance the algorithm's robustness, a few parameters are proposed for removing the wrong similar blocks. Experiment results demonstrate the ability of the proposed method to detect multiple examples of copy-move forgery and precisely locate the duplicated regions, even when dealing with images distorted by slight rotation and scaling, JPEG compression, blurring, and brightness adjustment.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

Nowadays, with the popularity of digital media cameras, digital media is playing an important role in our life. However, digital images can be manipulated and altered easily without leaving visible clues using digital image tools (e.g., Photoshop and 3D Max). This poses a serious social problem of the extent of trust that can be placed in the authenticity of digital content, especially when presented as evidence in a courtroom, for claiming insurance, and in the scientific world. According to some statistics [1], many journal-accepted manuscripts contain figures with inappropriate and fraudulent manipulations. Various methods have been developed to counter tampering and forgery for ensuring image authenticity [2].

Two approaches to ratifying the authenticity of a digital image can be categorized as active [3–5] and passive (blind) [6,7]. The active approach does not use actual digital content. Instead it focuses on implementing certain measures as the content is being created to confirm its authenticity afterward. Digital watermarking is the most popular approach in this category. Unlike the active approach, digital image forensics (also called passive image forensics) is a form of image analysis for finding out the condition of an image without the need for a priori information (such as embedded watermarks or signatures) and make a blind decision about whether the image has been tampered with. Most passive techniques are based on supervised learning through the extraction of specific features to distinguish the original images from tampered ones. The practicality and wide applicability of passive methods make them a popular research field. Copy-move forgery is one of the most commonly used forgery techniques that employ typical image processing tools (e.g., Photoshop and CorelDRAW). In copy-move forgery, a part of the image is copied and pasted in another part of the same image to conceal an object or to duplicate certain image elements. However, the task of detecting instances of forgery can be made significantly more difficult by post-processing on tampered images. Counterfeiters can use retouching tools, JPEG compression, or brightness to further alter forged images. To enhance the effects of image combination, the copied area can also be slightly rotated, scaled, or blurred to conceal the forgery. Thus, the effectiveness of copy-move forgery detection depends on the ability to detect regions of image duplication without being affected by post-processing operations, such as rotation, scaling, or JPEG compression. An example of copy-move forgery is shown in Fig. 1. The original image (Fig. 1(a)) has one bird, whereas the forged image (Fig. 1(b)) was manipulated using the cloning tool

☆ This paper has been recommended for acceptance by Prof. M.T. Sun.

http://dx.doi.org/10.1016/j.jvcir.2015.07.007
1047-3203/© 2015 Elsevier Inc. All rights reserved.

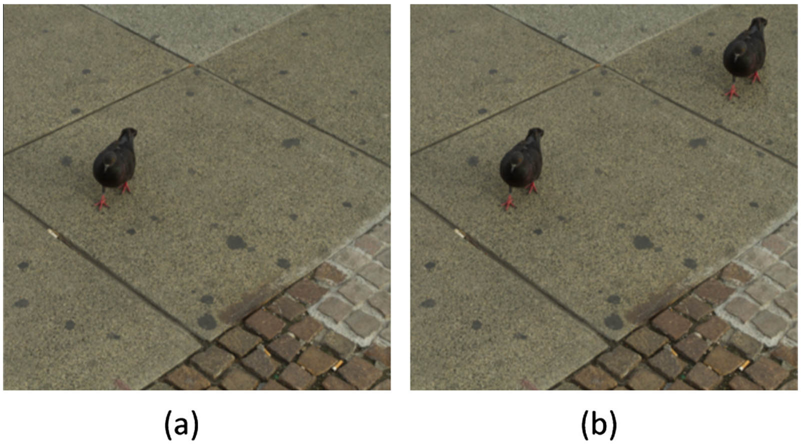

Fig. 1. Example of copy-move forgery (a) original image and (b) tampered image.

in Photoshop to show more than one bird by duplicating the bird present in the original image.

In this paper, we propose a scheme for detecting instances of copy-move forgery and authenticating images based on the Gabor transform [8]. The image is first converted into a gray-scale image and divided into overlapping fixed-size blocks. The proposed method, histogram of orientated Gabor magnitude (HOGM) descriptor, is then applied to each block for the extraction of local features and reduced the dimension of feature vectors to facilitate the measurement of similarity. Finally, each feature vector is lexicographically sorted, and regions of image forgery are detected through the identification of similar block pairs. A flow-chart of the proposed forgery detection method is shown in Fig. 2. In addition, we conducted rigorous experiments using images modified using highly convincing techniques to demonstrate the robustness of the proposed method in dealing with multiple copy-move forgeries. Compared with other methods, the main advantages of our method can be summarized as follows:

1. The histogram of orientated Gabor magnitude (HOGM) is proposed for the extraction of features from images that are suspected of forgery. Experiment results demonstrate the effectiveness of the proposed algorithm in detecting and precisely locating multiple instances of copy-move forgery within a single image. In addition, HOGM descriptors are well-suited for use in the analysis of image textures.
2. To reduce the probability of false matches, we developed a noise detector for the removal of false blocks.
3. The proposed technique is able to precisely locate regions of duplication without being affected by common post-processing techniques, such as image rotation, scaling, JPEG compression, blurring, and brightness adjustment. In most cases, the proposed method achieves better performance than other well-known approaches. In addition, the proposed method is even effective in dealing with images of high resolution.
4. Compared to most existing copy-move forgery detection techniques, the proposed method has a lower feature vector, which reduces computational complexity. Thus, this study makes a valuable contribution to the field of multimedia forensics.

The remainder of the paper is organized as follows. In Section 2, the related research about the past works is introduced, and Section 3 describes the Gabor filter. Section 4 gives the proposed method for detecting copy-move forgery. In Section 5, we present the results of experiments designed to evaluate the performance of the proposed method in terms of detection accuracy and computational complexity. The conclusions of this study are presented in Section 6.

2. Related research

Recently, many methods have been proposed to detect various forms of copy-move image forgeries. However, most methods used in the detection of image forgery can be categorized as either block-based methods or keypoint-based methods. The first such method was proposed by Fridrich et al. [6] by using a block matching detection scheme based on the discrete cosine transform (DCT). Popescu and Farid [9] proposed a copy-move forgery detection method that is different in the representation overlapping image blocks by using principal component analysis (PCA) instead of DCT. Luo et al. [10] divided blocks into four sub-blocks, which were evaluated according to the averages of the red, blue, and green color values. This method proved robust to some attacks such as JPEG compression, Gaussian blurring, and additive noise. Kang and Wei [11] applied singular value decomposition (SVD) to each image block to yield a representation with reduced dimensions, the feature matrix of which was then sorted lexicographically according to singular values. This approach achieved robust against noise distortion. Bayram et al. [12] applied the Fourier Mellin transform (FMT) and 1-D projection of log-polar values in a robust

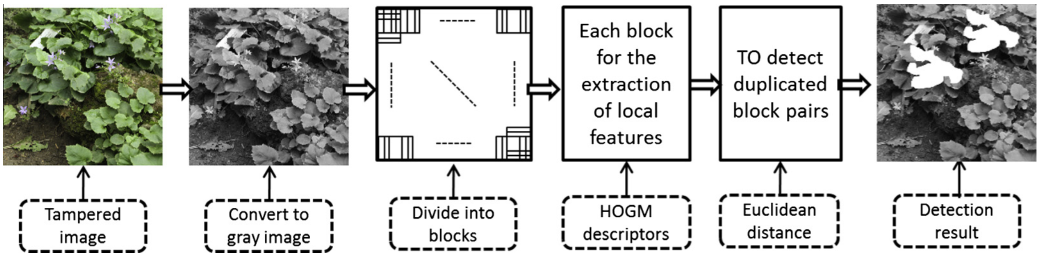

Fig. 2. The flow-chart of detection algorithm.

scheme for detecting of image forgery. Mahdian and Saic [13] exploited blur-invariant moments to detect duplicated regions, which provided robustness against post-processing such as blur degradation, additional noise, and arbitrary contrast changes. Li et al. [14] extracted features from circular blocks using rotation-invariant uniform local binary patterns (LBP). Lynch et al. [15] proposed an efficient expanding block algorithm based on direct block comparison rather than indirect comparisons based on block features. Zhao and Guo [16] applied DCT and SVD for the detection of image forgery. Lin et al. [17] proposed dividing each image into overlapping blocks of a fixed size, which are then represented using blocks with nine different average intensities before being undergoing radix sorting of feature vectors. Ryu et al. [18] employed Zernike moments in the extraction of features for block matching. They achieved an average detection precision of 83.59% in the case of region rotation. There is a growing number of cases in which copy-move forgery is masked using inpainting attacks, in which the filling of forged regions is based on information in the region surrounding the forgery. The objective behind inpainting attacks is to modify the content of an image in a visually plausible manner. Chang et al. [19] recently employed the computation of similar blocks to detect suspicious regions, whereupon forged regions were identified using the multi-region relation (MRR) technique. This method proved highly robust against typical copy-move and inpainting attacks. However, in cases where the forged region undergoes further processing using techniques such as scaling or rotation, this approach may fail. Recently, Liang et al. [20] presented an efficient forgery detection algorithm for object removal by exemplar-based inpainting. This method can reduce processing time while maintaining detection precision. Nearly all of these methods are based on a large number of blocks, such that the dimension of feature vectors is large and computational complexity is an issue. This increases computational complexity because multiple-index sorting is required to enable the lexicographical sorting of all blocks.

On the other hand, a few approaches involve extracting interest points (e.g., SIFT and SURF) from the entire image. Keypoint-based methods differ from block-based methods in their reliance on the identification and selection of high-entropy regions within an image (i.e., "keypoints"). A few approaches extract interest points by using a scale-invariant feature transform (SIFT) [21,22] that can detect and describe clusters of points belonging to cloned areas. For example, Amerini et al. [21] developed a SIFT-based method for detecting copy-move attacks and transformation recovery. Pan and Lyu [22] estimated the transform between matched SIFT keypoints and searched all pixels within duplicated regions after discounting the estimated transforms. SIFT keypoints guarantee geometric invariance; therefore, these methods enable the detection of rotated duplication. Nonetheless, SIFT-based schemes are still limited in their detection performance, due to the ability to extract keypoints only from a few specific locations in an image. That is, this method fails to detect copied and moved smooth areas, because the SIFT algorithm cannot extract features from those areas. In addition, keypoint-based methods are susceptible to a number of post-processing operations, such as blurring and flipping. Gilinsky and Manor [23] recently proposed the SIFTpack algorithm to enable the compact storage of sets of SIFT descriptors in order to reduce the storage requirements and running time. The SIFTpack method is an effective means of overcoming many of the drawbacks of SIFT-based schemes for the detection of image forgery. Shivakumar and Baboo [24] proposed a method for detecting copy-move forgery based on speeded up robust features (SURF), which can detect duplicated regions of various sizes with the minimum number of false matches when dealing with high-resolution images. Chen et al. [25] recently employed Harris corner interest points for extracting image keypoints, as well as step sector statistics for representing small circular image regions around each Harris point using a feature vector. Silva et al. [26] proposed a new method for copy-move forgery detection based on multi-scale analysis and voting processes of a digital image. This method extracts interest points robust against scaling and rotation forgeries and finds possible correspondences among them. Unfortunately, most keypoint-based methods only detect keypoints in an image. However, some keypoints in duplicate regions cannot be identified using keypoint-based algorithms. Furthermore, copied regions with little textural structure may be missed entirely. Based on the above analysis, both methods have strengths and weaknesses. Owing to differences in computational cost and performance, we have to consider the importance of the differences between block-based and keypoint-based methods.

The Gabor filter (or Gabor wavelet) has been widely adopted as an efficient means of extracting texture features for image retrieval [27,28]. Manjunath and Ma [27] proposed 48 Gabor wavelet features for browsing and retrieving image data. In [28], they applied this approach to the problem of texture analysis. Daugman [29] explains how the two-dimensional (2D) Gabor filter can attain the optimum resolution in terms of both space and frequency. This paper proposes a block-based framework that employs a histogram of orientated Gabor magnitude (HOGM) for extracting orientated features that can be used as evidence to demonstrate copy-move manipulation. We conducted rigorous experiments using images modified using highly convincing techniques in order to demonstrate the robustness of the proposed method in dealing with multiple copy-move forgeries. Finally, the proposed technique is able to precisely locate duplicated regions without being affected by common post-processing attacks, such as image small rotation, scaling, JPEG compression, blurring, and brightness adjustment.

Before moving to the process of detecting copy-move image forgery, we should discuss which image types were selected for this study. Most existing methods used in the detection of copy-move image forgery were designed for gray-scale images [6,9,13,15,17,22,25]. Different feature extraction methods require different types of images, which can have a strong influence on the performance of the proposed algorithm. A number of methods require that color images be converted to the YCbCr color system [6,10,14,16,18,24]. Regardless of image type, the luminance plane contains most of the geometric and visually significant information and the proposed HOGM method provides strong performance in defining local features on the luminance plane. Thus, we focused on the luminance component of the YCbCr space.

3. Review of Gabor filter

3.1. Gabor filter construction

Daugman [29,30] and Marcelja [31] modeled the responses of the visual cortex using Gabor functions because these functions are similar to the receptive field profiles of mammalian cortical simple cells. Gabor filters are extremely useful for texture analysis because of the 2D spectral specificity of texture, as well as textural variation with 2D spatial position. Daugman [29,30] developed the 2D Gabor functions (a series of local spatial band pass filters), which have good spatial localization, orientation selectivity, and frequency selectivity. In addition to accurate 2D space and 2D spatial frequency location, they provide robustness against varying image brightness and contrast as well [32]. The general form of a 2D Gabor filter is expressed as follows:

$$G_{\sigma,f,\theta}(x,y) = g_{\sigma}(x,y) \cdot \exp[2\pi jf(x\cos\theta + y\sin\theta)] \tag{1}$$

where

$$g_\sigma(x,y) = \frac{1}{2\pi\sigma^2}\exp[-(x^2+y^2)/2\sigma^2] \tag{2}$$

where $j = \sqrt{-1}$, f is the frequency of the sinusoidal wave, θ controls the orientation of the function, and $g_\sigma(x,y)$ is the Gaussian function with scale parameter σ. The parameters of the Gabor filter are therefore given by frequency f, orientation θ, and the scale σ. Note that we need to consider θ only in the $[0^\circ, 180^\circ]$ interval. Symmetry makes the other directions redundant. However, most 2D Gabor filters have slight response to regions of uniform luminance (or DC). This direct current (DC) response is zero for 2D Gabor filters having pure sine phase and the highest for filters having pure cosine phase. To eliminate sensitivity to illumination, we removed the DC of the 2D Gabor filter by applying the following formula:

$$Z_{\sigma,f,\theta}(x,y) = G_{\sigma,f,\theta}(x,y) - \frac{\sum_{i=-k}^{k}\sum_{j=-k}^{k} G_{\sigma,f,\theta}(x,y)}{(2k+1)^2} \tag{3}$$

where $(2k+1)^2$ is the filter size. In this paper, we express Eq. (3) in the complex form $Z_{\sigma,f,\theta}(x,y) = R_{\sigma,f,\theta}(x,y) + jI_{\sigma,f,\theta}(x,y)$. $R_{\sigma,f,\theta}(x,y)$ and $I_{\sigma,f,\theta}(x,y)$ are the real and imaginary parts, respectively, of the 2D Gabor filter. Such Gabor filters have been widely used in various applications [29–32]. In addition to accurate time-frequency location, they are robust against varying image brightness and contrast. Based on these properties, in this study, we can only rely on the shape and texture of the tampered images. Therefore, the Gabor filter is more appropriate.

3.2. Feature extraction with Gabor magnitude

Let $f(x,y)$ denote a grayscale image and $G_{\sigma,f,\theta}(x,y)$ represent a Gabor filter defined by its wave frequency f, scale parameter σ, and orientation θ. Given a neighborhood window of size $w \times w$ for $w = 2k+1$, discrete convolutions of $f(x, y)$ with the respective real and imaginary components of $G_{\sigma,f,\theta}(x,y)$ are

$$C_R(x,y)_{\sigma,f,\theta} = \sum_{X=-k}^{k}\sum_{Y=-k}^{k} f(x+X, y+Y)\cdot R_{\sigma,f,\theta}(X,Y) \tag{4}$$

$$C_I(x,y)_{\sigma,f,\theta} = \sum_{X=-k}^{k}\sum_{Y=-k}^{k} f(x+X, y+Y)\cdot I_{\sigma,f,\theta}(X,Y) \tag{5}$$

Based on these results, the magnitude responses $M(x,y)_{\sigma,f,\theta}$ of the Gabor filter can be computed as follow:

$$M(x,y)_{\sigma,f,\theta} = \sqrt{C_R^2(x,y)_{\sigma,f,\theta} + C_I^2(x,y)_{\sigma,f,\theta}} \tag{6}$$

The Gabor magnitude output of an image $f(x,y)$ is obtained by convolution of each block with the Gabor filter until the entire image is traversed. In this paper, the histogram of orientated Gabor magnitude (HOGM) is proposed for reflecting tampered images' features. After a forged image is divided into overlapping sub-blocks, sliding windows can be used to shift toward detected regions. Each window represents a region from the forged image, and the corresponding HOGM feature vector is calculated, which reflects magnitude information of the image region.

The HOGM method proposed herein focuses on the orientated of Gabor magnitude. The basic idea is that local object appearance and shape can often be characterized rather well based on the distribution of local magnitude directions. In most cases, one would use 2D Gabor filters with eight different orientations, $\theta \in \{0, \pi/8, 2\pi/8, \ldots, 7\pi/8\}$. To overcome the different orientations of texture information in the tampered images, we define m as the number of orientations for calculating $\theta_k = \pi(k-1)/m$, $k = 1, \ldots, m$. Therefore, there are m filters that have same frequency and scale parameter, but different orientations are selected for extracting magnitude information. In the proposed method, the HOGM scheme extracts the maximum magnitude values of the image from different orientations because the maximum magnitude values can show the orientation of the strongest textural information. The orientation of point (x,y) is defined as follows:

$$d(x,y) = \arg\max_{k=1,\ldots,m}\{M(x,y)_{\sigma,f,\theta_k}\} \tag{7}$$

where $d(x,y)$ is defined the orientation that corresponds to the maximum magnitude values. In this study, we consider 12 bins ($m = 12$) for the local histogram. The histogram channels are evenly spread over 0–180°, so each histogram bin corresponds to a 15° orientation interval.

In this study, HOGM is used to reflect tampered images' features. After a forged image is divided into overlapping sub-blocks ("cells"), the sliding window operation can be used to detect forged regions. Each window represents a region in the forged image, and the corresponding HOGM feature vector is calculated, which reflects the texture information of image region. For each cell, a local 1-D histogram of magnitude orientation over all pixels in the cell is accumulated. All histograms can be concatenated in a single feature vector, representing the HOGM descriptor. The procedure of HOGM feature vector calculation is described in Algorithm 1.

Algorithm 1 Procedure of HOGM feature vector calculation

```
HOGM (1: k) = 0;                    // k is the number of orientations
for i = 1 to n                      // n is the size of column
  for j = 1 to m                    // m is the size of row
    level = array(j, i);
    HOGM(level) = HOGM(level) + 1;
  end
end
```

This study used Algorithm 1 for the analysis of image texture. ***Array*** is defined as the image of Gabor magnitude that corresponds to the maximum magnitude values. In the ***array***, each pixel is obtained using Eq. (7). The image of Gabor magnitude is divided into overlapping cells of size $n \times m$. For each cell, the HOMG consists of the orientation levels of the magnitude; that is, it is a graph indicating the number of times each the orientation level of the magnitude occurs in the image. In algorithm 1, $k \in \{1, 2, 3, \ldots, 12\}$ and we compute 12 elements array **HOGM** (1: k). Each of the elements corresponds to one of the bins which contain the orientations of the pixels. Thus, each pixel is said to vote for one of the bins in the histogram. This vote is weighted by the magnitude orientation of the gradient at that pixel. The HOGM feature vector can be computed as follows:

$$\mathbf{HOGM}\ (1:k) = \sum_{i=1}^{n}\sum_{j=1}^{m}\begin{cases}1, & \mathit{if}\ \boldsymbol{array}(i,j) = k\\ 0, & \mathit{otherwise}\end{cases} \quad k \in \{1,2,3,\ldots,12\} \tag{8}$$

Fig. 3 presents an example of HOGM feature vector extraction. We selected two sub-blocks (Fig. 3(c) and (d)) from the image of Gabor magnitude (Fig. 3(b)) for the extraction of the HOGM feature vector, as shown in Fig. 3(e) and (f). Owing to visual effects, the size of the sub-block was set at 32 × 32 pixels for the extraction of HOGM descriptors. As shown in Fig. 3(e), most HOGM descriptors are concentrated in the 75–90° range. The orientations of image texture in Fig. 3(c) correspond to the HOGM descriptors in Fig. 3(e). Other experiments produced the same results, as shown in Fig. 3(d) and (f). HOGM descriptors are clearly able to capture local contour information, edge structures, and local shape characteristics, making them suited for use in the detection of copy-move image forgery.

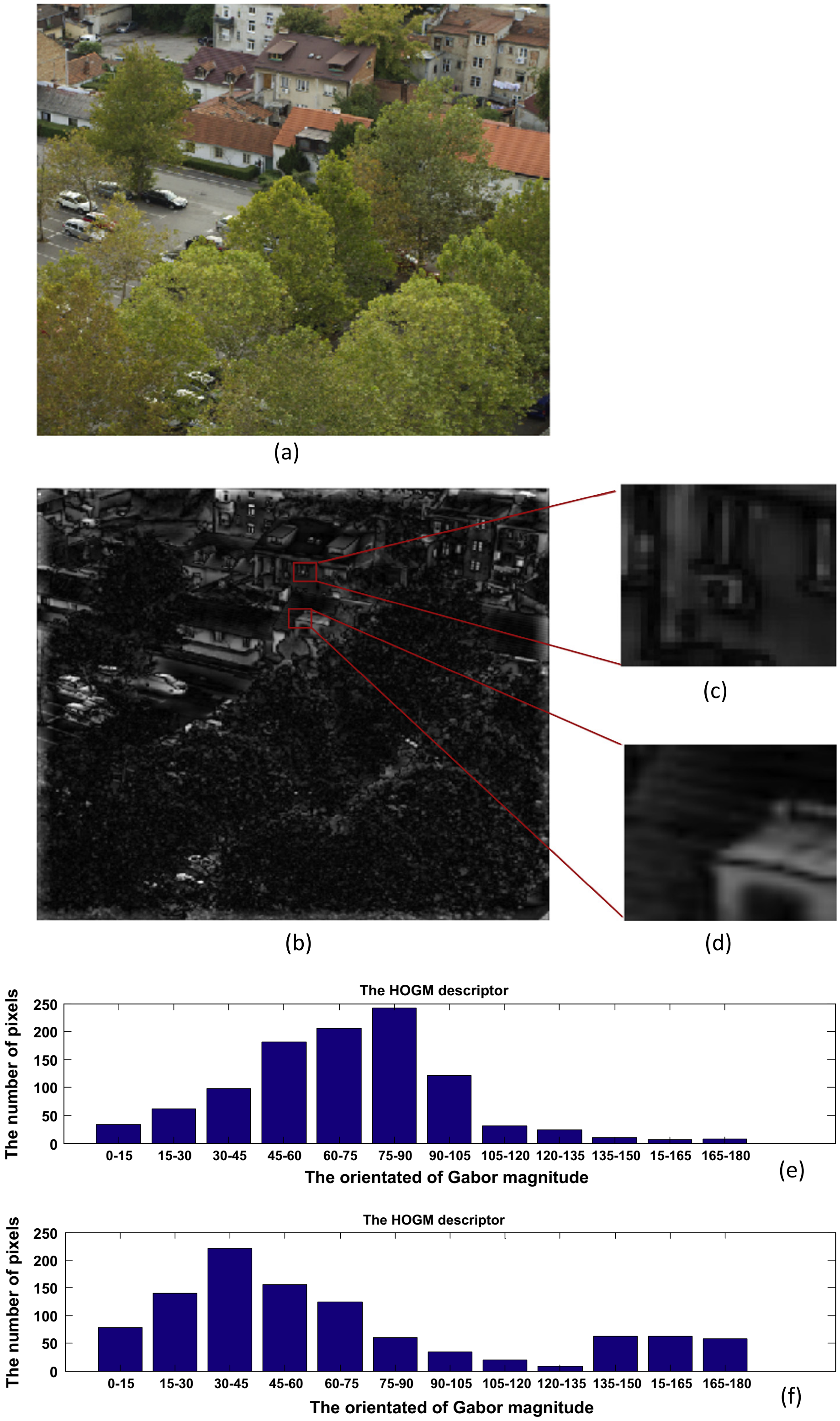

Fig. 3. Example of HOGM feature vector extraction: (a) original image; (b) image of Gabor magnitude; (c) and (d) sub-blocks of (b), & (e) and (f) are HOGM descriptors of (c) and (d).

4. Copy-move forgery detection scheme

The most important function of an algorithm in copy-move image forgery detection is determining whether a given image contains duplicated regions. If various post-processing operations such as rotation, blur degradation, and contrast changes carried out on copied regions are unknown, comparing every possible pair pixel by pixel to determine image forgery will raise the computational complexity to practically unmanageable level. Obviously, it is more practical to divide the suspicious image into blocks for detecting the duplicated regions.

In order to ensure efficient detection, some appropriate and robust features must be extracted from the blocks. These features should represent the entire block, provide robustness against common post-processing operations, and reduce computational complexity of the detection algorithm.

4.1. Proposed method

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

(1) Divide the suspicious image into overlapping blocks of a fixed size.
(2) Extract the features of each block using HOGM descriptors.
(3) Search for similar block pairs.
(4) Find the correct blocks and output them.

The entire architecture of the proposed forgery detection scheme is shown in Fig. 2.

4.2. Implementation details

In our method, we first divided the original image into overlapping blocks of a fixed size, detected the similarity of these blocks, and finally displayed the possible duplicated regions. The details of the proposed algorithm are as follows:

Step 1: Image pre-processing

Initially, the RGB image C is transformed into the gray scale image I, according to the following formula:

$$I = 0.299\mathrm{R} + 0.587\mathrm{G} + 0.114\mathrm{B} \tag{9}$$

where R, G, B denote red, green, and blue channels of the input color image, and I represents the luminance component of the YCbCr space. The YCbCr color space was included as a part of ITU-R BT.601 during the development of world-wide standards for digital video components. This formula reflects the fact that the human eye is more sensitive to some wavelengths of light than others, which can alter the perceived brightness of a given color. In addition, the luminance component of the YCbCr space contains more spatial information than do other color space environments. Thus, we employed the luminance component of the YCbCr space for the detection of copy-move forgery images.

Step 2: Dividing image into overlapping blocks of fixed size

To identify forged regions, the image is divided into overlapping square sub-blocks. The grayscale image I of $M \times N$ is first divided into overlapping sub-blocks of $B \times B$ for calculating HOGM descriptors. Each block is denoted as B_{ij}, where i and j indicate the starting points of the block's row and column, respectively.

$$B_{ij}(x,y) = I(x+j, y+i) \tag{10}$$

where x, $y \in \{0,\ldots,B-1\}$, $i \in \{1,\ldots,M-B+1\}$, and $j \in \{1,\ldots,N-B+1\}$. Hence, the image is then divided into $(M-B+1) \times (N-B+1)$ overlapping blocks.

Step 3: Obtaining HOGM descriptors of the same size

After applying HOGM to each block, an HOGM descriptor matrix with the same size as the block is assembled to represent each corresponding block. Here, we consider 12 bins for the local histogram. The resulting cell histograms are then combined into descriptor vectors for each block, such that 12 features can be used to represent each block. Thus, a $B \times B$ block is represented by a 1×12 feature vector $V = (x_1, x_2, \ldots, x_{12})$. The feature dimensions of this vector are lower than those used in other block-based detection methods [6,9,12,21].

Step 4: Comparison of HOGM features in each block

The feature vectors extracted in Step 3 are arranged in a matrix, denoted as A, of size $(M-B+1)(N-B+1) \times 12$.

$$A = \begin{bmatrix} V_1 \\ \vdots \\ V_{(M-B+1)(N-B+1)} \end{bmatrix} \tag{11}$$

However, detecting duplicated block pairs from A within a reasonable period of time can be exceedingly difficult. A brute-force search would be computationally very expensive; therefore, the authors in [6,9] proposed lexicographical sorting of feature vectors, such that similar features are found in different blocks. To reduce the required matching time, similar feature vectors are stored in neighboring rows. Thus, detection can be achieved through lexicographical sorting of the rows in matrix A, such that the features of duplicated block pairs appear successively. Because each element of A is a vector, the sorted set is defined as $\widehat{A}$. This study employed block matching to match corresponding blocks and identify regions that are likely to have been forged. Based on $\widehat{A}$, the Euclidean distance between adjacent pairs of $\widehat{A}$ is calculated. For accurate identification of the forged region, the distance threshold T_d and the similarity threshold T_s should be predetermined. There are distinct similarities in the feature vectors of blocks with overlapping pixels; therefore, we compared only blocks in which the position distance from other blocks exceeds distance threshold T_d. In this manner, we were able to calculate the actual distance between two similar blocks as follows:

$$B_{distance}(\widehat{V}_i, \widehat{V}_{i+j}) = \sqrt{(x_i - x_{i+j})^2 + (y_i - y_{i+j})^2} \tag{12}$$

where (x, y) is the center of the corresponding block.

In addition, block matching begins in the first row of matrix $\widehat{A}$. For a feature located in the ith row $\widehat{A}_i$, distances in the vicinity of ith rows are computed, and the smallest distance, denoted by $D(i)$, between the ith row and the nearby ith rows is obtained as follows:

$$D(i) = min\{D(i;i-j),\ldots,D(i;i-1), D(i;i+1),\ldots,D(i;i+j)\} \tag{13}$$

In this study, we set $j = 5$. If $D(i)$ is smaller than T_s, the corresponding blocks are regarded as being correctly matched, and the locations of the two blocks are stored. The matching process is repeated for all rows of $\widehat{A}$, and all matched block pairs are saved in set ω. Finally, we use Eqs. (12) and (13) to determine whether the blocks are duplicated. Fig. 4 shows the procedure of block matching.

Step 5: Post-processing of detection result

When all matched block pairs are saved in set ω, the forged regions can be identified by marking the copied and modified regions, and removing the isolated blocks. Generally, all detected blocks, including the original and forged blocks, are marked to generate the final detection result. Fig. 5 shows an example of the proposed marking method.

In most cases, the falsely fractional blocks marked on the initial detection map (using white pixels) should be removed. Thus, we designed a noise detector to facilitate the removal of these blocks. Suppose that the marked image (Fig. 5(c)) is divided into n non-overlapping blocks 16×16 pixels in size. The number of white pixels in each block is calculated. If the number of white pixels in a block is less than 64, then the block is treated as a false block and returned to the original gray-scale image. Otherwise, the white pixels are retained and no further action is taken.

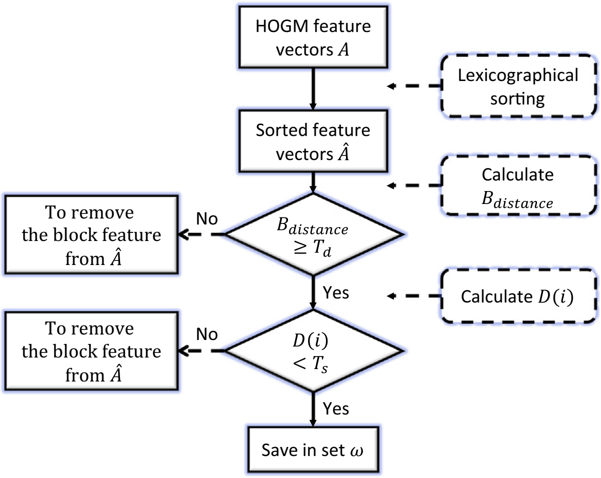

Fig. 4. Procedure of block matching.

Following the detection process, this leads to the removal of a small number of isolated false matches. Fig. 5(d) presents the detection results obtained after applying the proposed detector.

4.3. Computational complexity in matching procedures

One additional problem in the detection of copy-move forgery is the computational complexity associated with block matching. In our experiment, forged regions are detected by lexicographical sorting of the image blocks, which depends primarily on the number of blocks; i.e., number of rows in matrix A. However, the number of feature vectors is also strongly associated with computation speed. Thus, the best case for lexicographical sorting requires a $O(\rho\alpha log_2\alpha)$ comparison operation for the sorting of matrix A, where α is the number of overlapping blocks and ρ is the number of feature vectors. After sorting matrix A, the sorted set is defined as $\hat{A}$. In Eq. (13), we employ Euclidean distance to determine whether blocks have been duplicated. The matching process is repeated for all rows of $\hat{A}$. Thus, the complexity of the proposed method requires only $O(2j\rho\alpha)$, where α, ρ are described earlier, and j is the number of nearby ith rows. In this study, we set $j = 5$. Therefore, the total computational complexity of the matching procedure is approximately $O(\rho\alpha log_2\alpha) + O(2j\rho\alpha)$. It should be noted that the number of overlapping blocks is obtained by $(M - B + 1) \times (N - B + 1)$ for a given image with $M \times N$ pixels; therefore, as the size of the image grows, the total number of sub-blocks increases (with $O(M \times N)$ complexity), which also affects the overall computational complexity. Clearly, the problem of computational complexity in the detection algorithm is caused by the number of matching blocks as well as the number of dimensions in the feature vector.

5. Experimental results and discussion

In this section, we describe the experimental results of our method. In our experiments, the proposed method is evaluated using two publicly available databases designed for image forgery detection. The first one was the CoMoFoD database [33]. All images

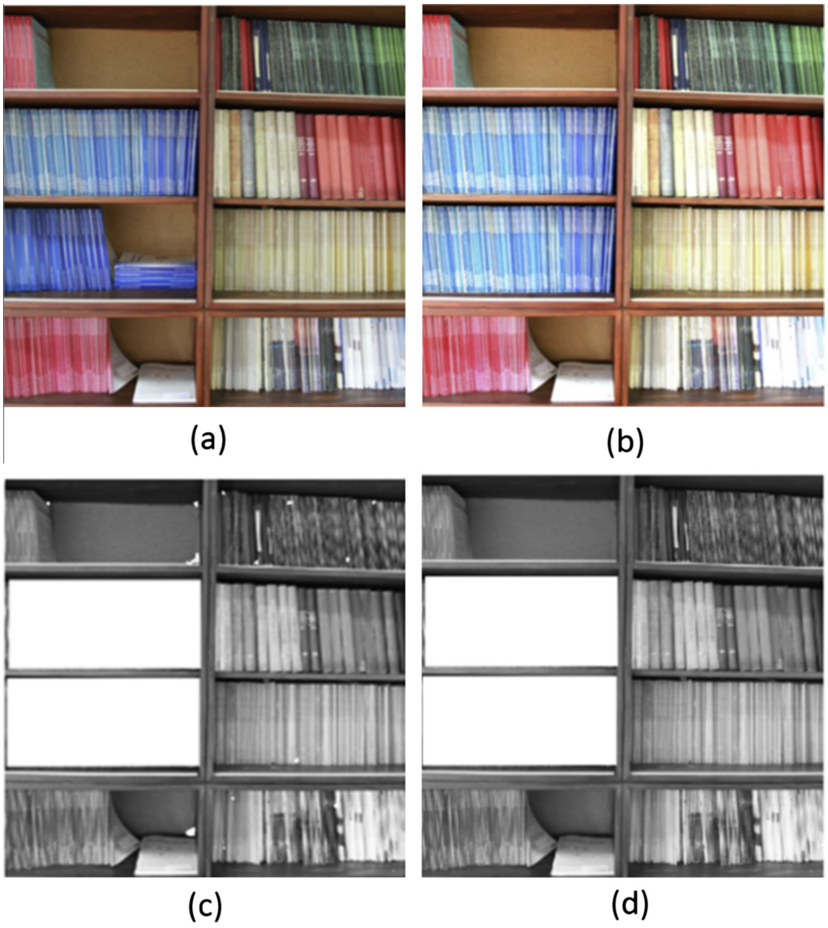

Fig. 5. Detection results: (a) original image, (b) tampered image, (c) initial detection result, and (d) final detection map.

were recorded by a Canon EOS 7D camera and stored in the CR2 (Canon RAW version 2) format as minimally processed data. These forgery images consist of 200 png images with a resolution of 512 × 512 pixels in the small image category. Data acquisition was performed under various outdoor conditions, such as a natural setting, among buildings, and overlooking the city. We selected 120 images from the CoMoFoD database, according to applied manipulation: translation (No. 001-040), rotation (No. 041-080) and scaling (No. 081-120). Herein, translation refers to the simple copying, moving, and pasting of a forged region to a new location within the same image, without performing any processes to distort the image. However, various post-processing methods, such as JPEG compression, blurring, and adjustment of brightness, are applied to all forged and original images. Moreover, the second dataset is comprised of several color PNG images released from the Image Manipulation Dataset [34]. All images are high-resolution (about 800 × 500 to 3200 × 2400 pixels), including 48 base images, separate snippets from these images, and there is a software framework for creating ground truth data. These images were manipulated using copy-move forgery in conjunction with other processes, such as scaling, rotation, JPEG compression, and combinations of these methods. The Image Manipulation Dataset includes realistic copy-move forgeries of high-resolution images obtained from consumer cameras. In the ground truth database, an average of approximately 10% of the pixels is in regions that have been tampered with. Fig. 6 presents the forged images used in the experiments. All experiments were performed on a personal computer with a 2.1 GHz CPU, 4 GB memory, under MatLab environment. The experimental results are presented in the following sections, according to the various processes used for manipulating the forged regions.

5.1. Performance evaluation

To illustrate the performance of the proposed algorithm, we referenced two evaluation criteria, i.e., correct detection ratio (CDR) and false detection ratio (FDR), which are defined as follows:

$$\mathrm{CDR} = \frac{|C \cap \widetilde{C}| + |F \cap \widetilde{F}|}{|C| + |F|} \tag{14}$$

$$\mathrm{FDR} = \frac{|\widetilde{C} - C| + |\widetilde{F} - F|}{|\widetilde{C}| + |\widetilde{F}|} \tag{15}$$

where C is the copy region, F is the tampered region, and $\widetilde{C}$ and $\widetilde{F}$ are the tampered copy region and the detected tampered region, respectively. $|\,|$ refers to the area of the region, $\cap$ refers to the intersection of two regions, and $-$ refers to the difference between two regions. CDR indicates the performance of the algorithm in terms of accurately locating the pixels of copy-move regions in the tampered image, while FDR reflects the percentage of pixels that are not contained in the duplicated region but are nevertheless included by the implemented method. In other words, the two metrics indicate the precision with which the proposed algorithm locates copy-move regions. The closer CDR to 1 and FDR to 0, the higher is the method's precision. In the following experiment, we selected over 2000 forged images from the two datasets (described in Section 5) to test the effectiveness of our algorithm. To evaluate the degree to which the size of the duplicated regions influences detection, we used sub-blocks of three sizes (16 × 16 pixels, 32 × 32 pixels, and 48 × 48 pixels) in the experiments.

5.2. Effectiveness and accuracy test

In this experiment, the effectiveness of our algorithm was evaluated by selecting 40 source images measuring 512 × 512 pixels from the first dataset as well as 48 source high-resolution images from the second dataset. All doctored images in this experiment are devoid of post-processing, and the corresponding detection results are shown in Fig. 7. The top row shows the tampered images, and the bottom row shows the detection results. Owing to space constraints, only a few of the experimental results are shown here.

Fig. 7 shows that the CDR is generally greater than 0.95 and the FDR equals 0, that is, our algorithm can locate tampered image regions quite precisely. In addition, Fig. 7 indicates that our algorithm can find duplicated regions precisely when all duplicated regions are non-regular and meaning, even though there are extremely similar scenes or flat regions in the image, such as large areas of sand or leaf. Owing to the homogenous backgrounds of the suspicious images, it is challenging to discern forgery. Images shown in Fig. 7 are the test results, which demonstrate that our algorithm works well even when the tampered images have multiple duplicated regions. However, other studies [6,9–12] have not considered such forgery.

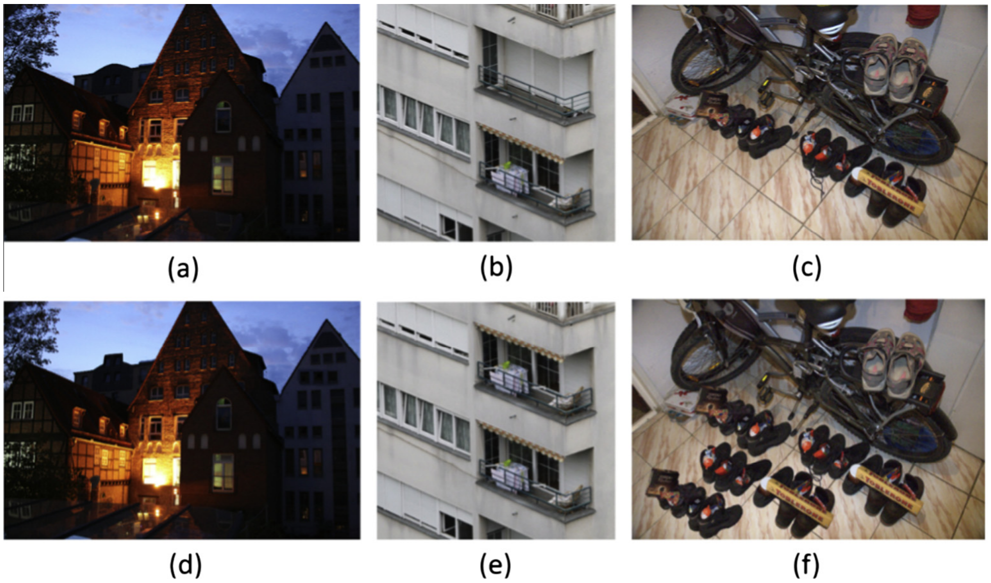

Fig. 6. Examples of copy-move forgery images: (a)–(c) are the original images and (d)–(f) are the tampered images.

The statistical detection rates with noise detectors for sub-blocks of various sizes are presented in Table 1, in which the CDR is generally greater than 0.9 and the FDR is nearly 0. The CoMoFoD database includes 40 images that were forged using copy-move techniques (No. 001_F∼040_F). Each of the forged images in the CoMoFoD database includes one or two regions of forgery. We also used 48 examples of copy-move forgery from the Image Manipulation Dataset to which no post-processing was applied, with the objective of determining which size of sub-block is best suited to evaluating the performance of the algorithm. According to Table 1, the proposed method performs well in the detection of duplications with block sizes of 16 × 16, and 32 × 32, but not as well for a block size of 48 × 48. This is because some portions of the forged regions are so small that they cannot be detected when using larger block sizes. As long as the tampered regions are larger than the blocks employed, the proposed algorithm can detect forgery with a high degree of precision. Thus, we fixed the block size at 16 × 16 for feature extraction. In addition, Table 1 also shows the effect of the addition or removal of the noise detector to the final detection results. It can be seen that the noise detector plays an important role in the authentication of images.

Table 1
Copy-move forgery detection results with/without noise detector.

Block size		16 × 16	32 × 32	48 × 48	Average
Detection results with noise detector					
CoMoFoD dataset	CDR	0.988	0.967	0.912	0.954
	FDR	0.028	0.045	0.124	0.066
Image Manipulation Dataset	CDR	0.997	0.986	0.939	0.974
	FDR	0.003	0.013	0.091	0.036
Detection results without noise detector					
CoMoFoD dataset	CDR	0.852	0.819	0.742	0.804
	FDR	0.224	0.232	0.277	0.244
Image Manipulation Dataset	CDR	0.874	0.854	0.809	0.845
	FDR	0.175	0.215	0.227	0.205

5.3. Robustness against post-processing

The ability to resist post-processing attacks is fundamental to copy-move forgery detection methods. Many types of post-processing attacks can be applied to forged images for hiding traces of tampering. The most common post-processing attacks are image rotation, scaling, JPEG compression, blurring, brightness adjustment, and combination. In addition to these common post-processing attacks, this paper made an attempt to defend against inpainting attacks [35,36]. In the following, we present details of the experiment results in which a variety of post-processing attacks were applied to duplicated regions.

5.3.1. Rotation attack

In this section, we examine cases in which the forged region is copied, rotated and moved to another position in the same image without distorting it using any other techniques. Experiments were conducted using images from the two databases in order to test the robustness of the proposed algorithm. Fig. 8 presents an example of rotation duplication forgery with the corresponding detection results. In the CoMoFoD database (No. 041_F∼080_F), each of the 40 images with rotation includes one or two regions of forgery. The duplicated regions are rotated by angles randomly selected between 1° and 180°. However, in the Image Manipulation Dataset, the duplicated regions with rotation attack are divided into two categories: small rotation angles between 2° and 10°, in steps of 2°, and large rotation angles of 20°, 60°, and 180°. Thus, each of the forged images was subjected to eight different rotation angles. The database includes 384 (48 × 8) forged images with rotation.

The CDR and FDR are listed in Table 2. The performance of the proposed method tends to decrease when dealing with duplicated regions rotated more than 6°. This can be attributed to differences in the HOGM descriptors for the original regions and the forged regions that have undergone large rotation. Nonetheless, the proposed method still provides good detection performance when used for images with small rotations. Most tampered images in actual situations undergo only slight rotation.

5.3.2. Scaling attack

In this section, we evaluate the robustness of the proposed scheme against scaling-duplication forgery. In this type of image, the forged region is copied, scaled and moved to another position

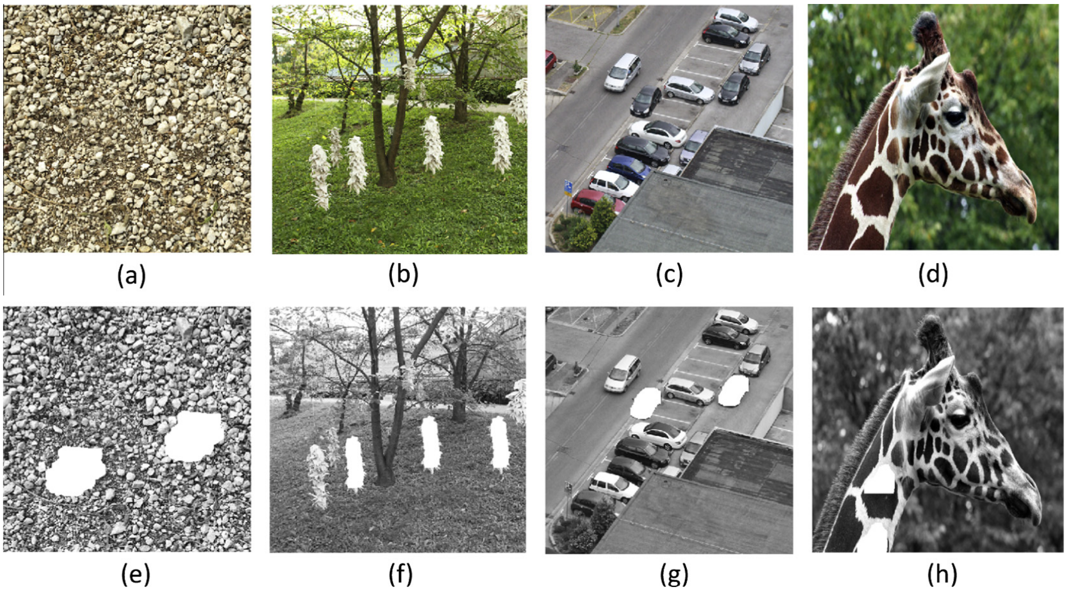

Fig. 7. Multiple copy-move image forgery detection: (a)–(d) are tampered images, (e)–(h) are the detection results of tampered images, and CDR/FDR rates are 0.974/0, 0.992/0, 0.982/0, and 0.998/0, respectively.

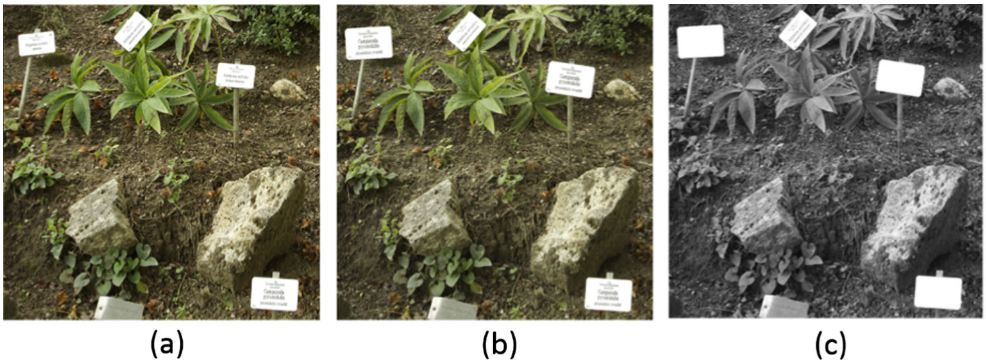

Fig. 8. Detection results using images that have undergone distortion by rotation: (a) original image (No. 054_O), (b) copied region rotated 5° and 7° (No. 054_F), and (c) final detection map.

in the same image without applying any additional forms of distortion. The experiments were conducted using two image databases. The CoMoFoD database includes 40 forged images with scaling (No. 081_F∼120_F). The forged regions were copied and then scaled up or down using random scaling factors between 0.4 and 1.5. Images in the Image Manipulation Dataset had forged regions rescaled to between 0.91 and 1.09 of their original size, in steps of 0.02. This database also provides images rescaled by 0.50, 0.8, 1.2 and 2.0 in order to test the degree to which the performance of algorithms degrades when the forged regions are subjected to large-scale resizing. In the dataset, each forged image was subjected to 14 different scaling, which resulted in 672 (48 × 14) images for evaluation.

Table 3 presents statistics related to the detection rates of scaling duplication using the two databases. Statistically, the detection results when dealing with scaling duplication are quite close to those of rotation duplication. The performance of the algorithm dropped when using a block size of 16 × 16 and scaling factor below 0.9 and above 1.1. Nonetheless, the proposed method performs very effectively when dealing with a scaling factor between 0.95 and 1.05. These results demonstrate the efficacy of the proposed method in the detection of duplication in conjunction with slight scaling.

5.3.3. JPGE compression attack

The fact that most forged images are stored in JPEG format means that the proposed image forgery detection algorithm must be able to deal with common disturbances associated with this form of compression. We employed the same two datasets as in the previous experiment to evaluate the robustness of the proposed algorithm to JPEG compression with various quality factors (between 20 and 100, in steps of 10). Image resolution was not altered when the images were saved. This resulted in 360 (40 × 9) forged images from CoMoFoD (No. 001_F_JC1∼040_F_JC9) and 432 forged images (48 × 9) from the Image Manipulation Dataset.

Experiments were performed to evaluate the effectiveness of the proposed algorithm in resisting the effects of JPEG compression. As shown in Table 4, the proposed method works, regardless of the quality factor applied during compression (Q = 20, 30, 40, 50, 60, 70, 80 and 90). Even when the forged images are compressed using a quality factor of only 60, the CDR value exceeds 0.9. In most cases, the FDRs remained below 0.1, which indicates that the results are promising. The proposed method is even able to detect JPEG compression with a quality factor above 40, resulting in acceptable CDR and FDR values.

5.3.4. Blurring and brightness change attacks

Rotation, scaling, and JPEG compression are commonly used to mask image copy-move forgery. However, forged images may also be subjected to other post-processing methods, such as blurring and brightness adjustment. In this section, we outline experiments aimed at evaluating the ability of the proposed algorithm in resisting blurring and changes in brightness. The Image Manipulation Dataset does not include forged images with blurring and/or brightness changes; therefore, only the CoMoFoD database was employed in these experiments. This included 120 forged images (No. 001_F_IB1∼040_F_IB3), which were blurred by convolving using three different averaging filters (3 × 3, 5 × 5 or 7 × 7). This resulted in a noticeable alteration of the images, particularly when using a 7 × 7 averaging filter. Fig. 9 presents an example of a forged image that underwent blurring with its corresponding detection results. We also evaluated the effects of changes in brightness using 120 forged images from the CoMoFoD database (No. 001_F_BC1∼040_F_BC3). Brightness was altered by mapping the intensity values of the original image between lower and upper bound at intervals [0, 1]. This resulted in images in three ranges of brightness ([0.01, 0.95], [0.01, 0.9] and [0.01, 0.8]). The effect of altering image brightness by [0.01, 0.95] was imperceptible. Altering brightness by [0.01, 0.8] presented a noticeable change in the appearance of the images. Fig. 10 presents an example of a forged image in which the brightness was altered by [0.01, 0.8] as well as the corresponding detection results.

As shown in Table 5, the proposed algorithm achieved high correct detection ratios for blurring and brightness adjustment. High detection performance was achieved when the images were distorted using 3 × 3 and 5 × 5 averaging filters; however, detection performance dropped when a 7 × 7 averaging filter was applied. As indicated by the CDR and FDR values in Table 5, the proposed

Table 2
Detection results using forged images distorted by rotation.

Rotation angles		2°	4°	6°	8°	10°	20°	60°	180°	Average
CoMoFoD dataset	CDR	0.93	0.88	0.83	0.69	0.55	0.12	0.09	0.35	0.55
	FDR	0.09	0.17	0.22	0.48	0.69	0.97	0.98	0.83	0.55
Image Manipulation Dataset	CDR	0.94	0.91	0.85	0.73	0.57	0.37	0.11	0.37	0.61
	FDR	0.10	0.12	0.21	0.42	0.64	0.79	0.96	0.77	0.50

Table 3
Detection results using forged images distorted by scaling.

Scaling factor		Below 0.90	0.91–0.94	0.95–0.99	1.01–1.05	1.06–1.09	Above 1.10	Average
CoMoFoD dataset	CDR	0.42	0.73	0.93	0.94	0.77	0.45	0.71
	FDR	0.72	0.33	0.06	0.08	0.25	0.62	0.34
Image Manipulation Dataset	CDR	0.45	0.75	0.93	0.95	0.79	0.52	0.73
	FDR	0.78	0.28	0.05	0.06	0.19	0.59	0.33

Table 4
Detection results of forged images distorted by JPEG compression.

Quality factor		90	80	70	60	50	40	30	20	Average
CoMoFoD dataset	CDR	0.97	0.96	0.92	0.90	0.84	0.69	0.52	0.32	0.76
	FDR	0.04	0.07	0.10	0.15	0.19	0.25	0.41	0.55	0.22
Image Manipulation Dataset	CDR	0.98	0.96	0.94	0.90	0.87	0.74	0.58	0.43	0.80
	FDR	0.03	0.05	0.07	0.11	0.12	0.19	0.37	0.45	0.17

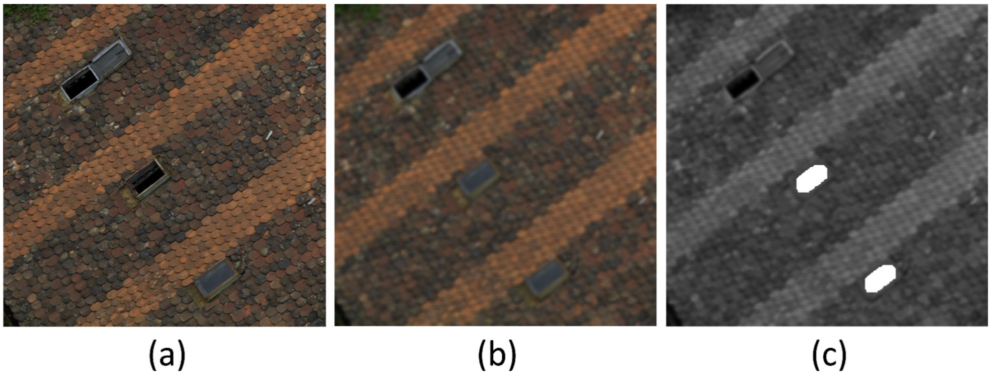

Fig. 9. Detection results using images that have undergone distortion by blurring: (a) original image, (b) image altered using 7 × 7 averaging filter (No. 021_F_IB3), and (c) final detection map.

algorithm also provides excellent robustness against changes in image brightness, as evidenced by the reliable detection performance achieved in the [0.01, 0.8] range.

5.3.5. Inpainting attack

To evaluate the robustness of the proposed method against different attacks, we designed an experiment involving an inpainting attack. Image inpainting is an actively growing field of research because it can effectively repair damaged or removed regions in a visually plausible way. All of the sample images were obtained from public online databases (http://www.dtic.upf.edu/~mbertalmio/restoration0.html). In this experiment, two common evaluation criteria (CDR/FDR) were not applicable because the copied regions of inpainted images are unknown. Fig. 11 presents an example of our detection results. Fig. 11(a) and (b) are the original images, Fig. 11(c) and (d) are the inpainted images, and Fig. 11(e) and (f) presents the final detection results. It is obvious that the proposed approach only detected a small portion of the inpainting. This illustrates that the proposed method is not robust against inpainting attacks.

5.4. Comparison with existing approaches

In this experiment, we compared the proposed method with other well-known approaches: DCT-based [6], PCA-based [9], FMT-based [12], and SIFT-based [21] as well as the methods

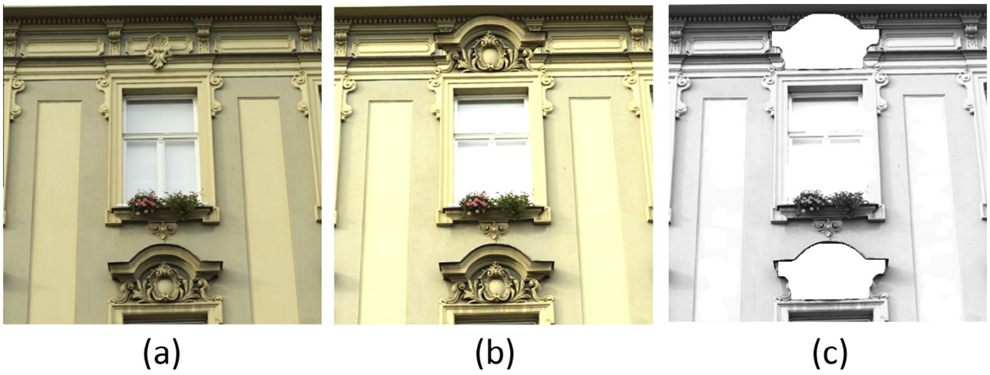

Fig. 10. Detection results using images that have undergone distortion by adjustment of brightness: (a) original image, (b) image with brightness altered [0.01, 0.8] (No. 034_F_BC3), and (c) final detection map.

Table 5
CDR and FDR using images from CoMoFoD database.

Attacks		CDR	FDR
Image blurring (filter size)	3 × 3	0.982	0.023
	5 × 5	0.976	0.052
	7 × 7	0.946	0.078
Average		0.968	0.051
Brightness adjustment (ranges)	[0.01, 0.95]	0.986	0.024
	[0.01, 0.9]	0.975	0.036
	[0.01, 0.8]	0.953	0.043
Average		0.971	0.034

proposed by Zernike [18] and Lin et al. [17]. Herein, I gratefully acknowledge technical support from [37]. Some of the source-code and datasets [34] are freely available on the internet. We selected more than 2000 forged images from two databases (the CoMoFoD dataset and Image Manipulation Dataset) to assess the performance of these methods. These tampered images were processed using rotation, scaling, and JPEG compression.

Fig. 12 presents a comparison of the overall simulation results of the forged images. Fig. 12(a) and (b) reveals the high detection performance of the proposed method when examining images distorted by JPEG compression up to JPEG quality level 50. This can be attributed to the fact that HOGM descriptors are able to capture the maximum magnitude values of the image from various spatial-frequencies. However, the maximum magnitude values indicate the orientation of the strongest textural information, such that HOGM descriptors present an accurate reflection of the main textural patterns in an image, thereby ensuring robust performance. In the case of JPEG compression, the tampered images were converted using various quality factors (30, 40, 50, 60, 70, 80, and 90). As shown in Fig. 12(a) and (b), regardless of method, an increase in the JPEG quality factor led to an increase in CDR and a decrease in FDR. The CDR/FDR curves show that the FMT-based [12] method and the method proposed by Lin et al. [17] were unable to equal the performance of the other methods. In particular, when the quality factor was less than 70, the CDR of the FMT-based method was distinctly below that of the other methods. Even at a JPEG quality factor of 50, the proposed method achieved a higher CDR than did the other methods. In most practical situations, the JPEG quality factor is generally higher than 50, making the proposed method highly effective in the handling of JPEG compression.

As shown in Fig. 12(c) and (d), similar results were observed in the case where forged regions underwent rotation using six different angles (2°, 4°, 6°, 8°, 10°, and 20°). The CDR/FDR curves in Fig. 12(c) and (d) show that the proposed method outperformed the other methods. As shown in the CDR curves in Fig. 12(c), the proposed method achieved good performance (CDR $\geqslant 0.8$) in cases where rotation was less than 6°. The CDR curves of the DCT-, PCA-, and FMT-based methods as well as the method proposed by Lin et al. [17] dropped rapidly when the angle of rotation was increased. Moreover, the FDR curves in Fig. 12(d) indicate that the proposed method is unable to deal with forged regions that have undergone a bigger angle of rotation. This can be attributed to the fact that we used the histogram of magnitude orientation

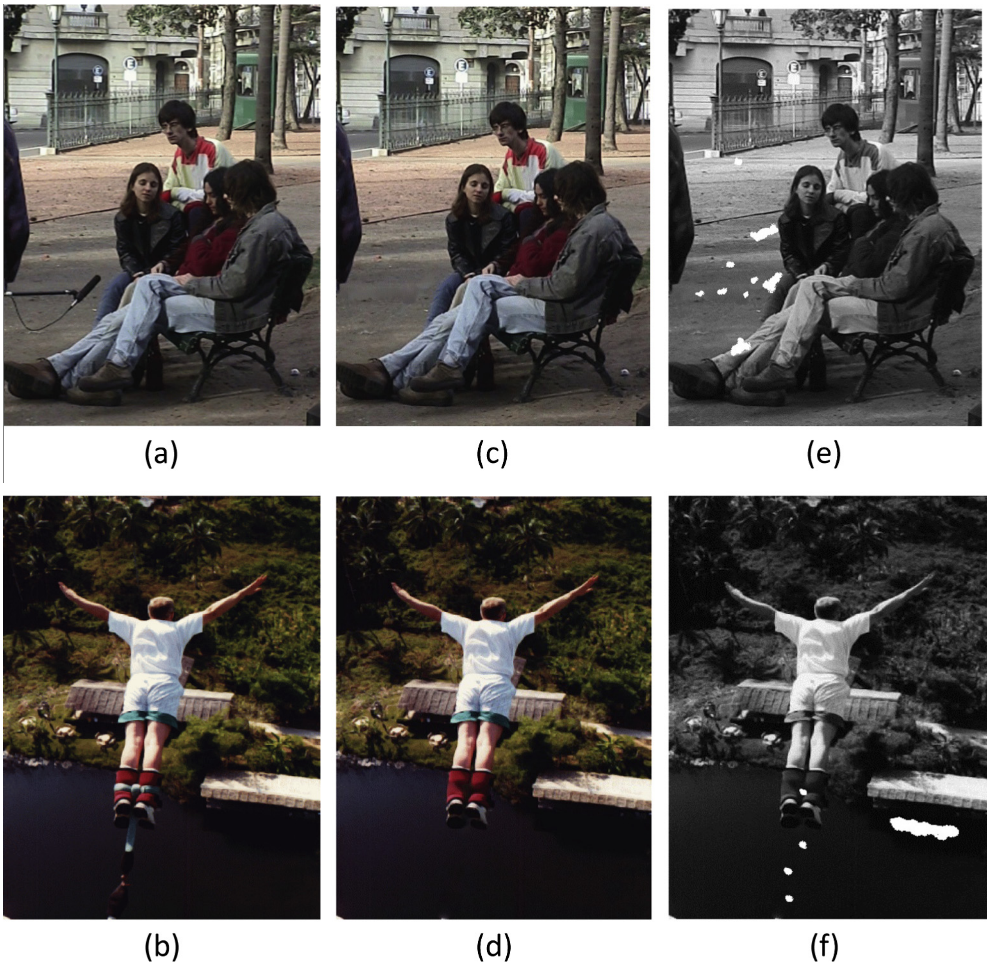

Fig. 11. Detection results using inpainting images (a) and (b) original image, (c) and (d) inpainting image, & (e) and (f) final detection map.

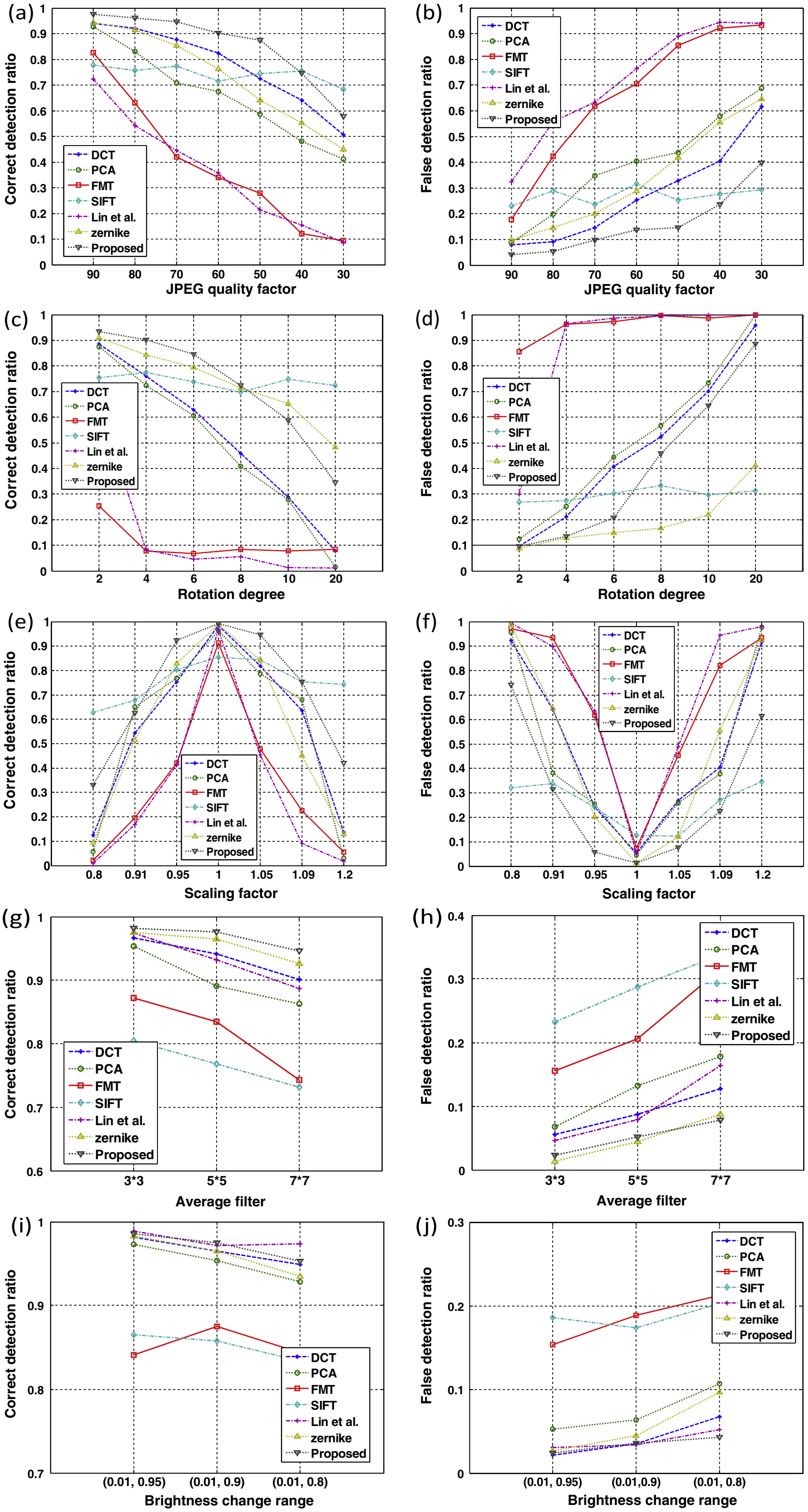

Fig. 12. CDR/FDR curves obtained using various methods: (a) and (b) various JPEG quality factors, (c) and (d) various angles of rotation, (e) and (f) various scaling factors, (g) and (h) various averaging filters, & (i) and (j) various brightness change ranges.

Table 6
Comparison of forgery-detection approaches.

Methods	Number of blocks	Feature dimensions	Feature extraction	Feature matching
PCA	255,025	64	31.792	44.841
DCT	255,025	32	476.874	40.281
FMT	247,009	45	58.266	41.502
Zernike	247,009	12	40.821	38.388
Lin et al.	247,009	9	3.017	37.303
SIFT	Approximately 2700 key-points	128	4.357	1.235
Proposed	247,009	12	36.771	38.472

to represent the characteristics of the forged region. In this study, each histogram bin corresponds to a 15° orientation interval, such that the HOGM descriptors are able to resist only forged regions with a slight rotation. In addition, the SIFT-based and Zernike methods enable the detection of rotated duplication because SIFT keypoints and Zernike moments guarantee geometric invariance. The other methods, particularly the FMT-based [12] method and the method proposed by Lin et al. [17] performed poorly when presented with images with larger rotations. The proposed method outperformed the SIFT-based and Zernike methods when the forged region was rotated only slightly, due to the fact that the HOGM descriptors are based on the shape and texture features of forged images. Detection results are commonly affected by larger rotations. Fig. 12(e) and (f) presents the results of a comparison between forged images that were altered using various scaling factors (0.8, 0.91, 0.95, 1, 1.05, 1.09, and 1.2), thereby illustrating the efficacy of the proposed method. As evidenced by the CDR/FDR curves, the proposed method is acceptable for slight scaling; however, it achieved the worst performance with a scaling factor above 1.09 and below 0.91. Overall, the SIFT-based method attained the best performance when the forged images with different scaling factors.

In the case of image blurring, Fig. 12(g) and (h) illustrates that the CDR curve of the proposed method outperforms other methods, with CDR $\geqslant 90\%$ when the dimensions of the averaging filter are reduced. With regard to FDR curves, the proposed method provided the lower FDR, even when using a 7×7 averaging filter. Fig. 12(i) and (j) presents comparison results for tempered images with the following changes in brightness: ([0.01, 0.95], [0.01, 0.9] and [0.01, 0.8]), illustrating the efficacy of the proposed method. Increasing the range of brightness resulted in an increase in CDR and decrease in FDR for all methods. CDR/FDR curves demonstrate that the SIFT-based and FMT-based methods have the worst performance, particularly in a brightness range of [0.01, 0.8]. On the same images, the proposed method achieved the higher CDR and lower FDR. As demonstrated in the above analysis, the proposed method may lead to a number of false matches; however, when combined with human interpretation, it can be very useful in confirming suspicions related to the authenticity of an image.

One additional problem in copy-move forgery detection is the computational complexity associated with block matching. In this experiment, we used a forged image with a resolution of 512×512 for evaluation. Table 6 presents a comparison of the proposed approach with other existing methods. The lexicographic sorting matrix is the major contributor to computational complexity. The total number of rows denotes the block number, and the total number of columns denotes the feature dimensions. The features in the proposed algorithm are 12-dimensional, while those in the other methods are as follows: DCT-based (64-dimensional) [6], PCA- (32-dimensional) [9], FMT-based (45-dimensional) [12], Zernike (12-dimensional) [18] and Lin et al. (9-dimensional) [17]. As can be inferred from Table 6, the lexicographic sorting matrix in the proposed algorithm is smaller than those used in the DCT- [6], PCA- [9], FMT- [12] based approaches under the same experiment conditions. Nonetheless, the proposed HOGM-descriptor-based approach provides superior detection efficiency.

In addition, the computation time depends on the number of the feature dimensions. Table 6 also shows the average run time in seconds over the CoMoFoD dataset for feature extraction and feature matching. Among the seven methods, the block-based methods [6,9,12,18,17] are very similar in terms of matching time, but the run times of feature extraction differ. The keypoint-based method (SIFT) exhibits outstanding performance in terms of run time for both feature extraction and matching. However, the feature sizes of SIFT descriptors are relatively large because the number of keypoints is smaller than the number of image blocks. This is the cause of the superiority of this method over block-based methods in terms of computation time. In the proposed method, the whole process takes only about 1.5 min, which is fast enough for image authentication. If the program code were optimized, it is likely that this time would be further reduced.

6. Conclusions

Copy-move is a common method for image forgery. It works without any digital watermarks or signature information. This paper proposes an effective method for detecting duplicated regions based on the histogram of Gabor magnitude. Compared with six existing methods, the features for the representation of blocks in the proposed approach are of lower dimensionality. Our experiment results demonstrate the efficacy of the proposed algorithm in detecting multiple instances copy-move forgery, while remaining robust against actions aimed at concealing forgery, including slight rotation, slight scaling, JPEG compression, blurring, and brightness adjustment. Furthermore, the computational complexity of the proposed algorithm is relatively low. This study, therefore, makes a valuable contribution to the field of multimedia forensics.

Nonetheless, image alteration can be concealed by methods of greater sophistication, such as great rotation, scaling, noise addition, inpainting or a combination thereof. This makes the detection of copy-move forgery far more challenging. We are currently developing methods to overcome these limitations.

References

[1] H. Farid, Exposing digital forgeries in scientific images, in: Presented at the Proceedings of the 8th Workshop on Multimedia and Security, Geneva, Switzerland, 2006.

[2] T. Gloe, M. Kirchner, A. Winkler, R. Behme, Can we trust digital image forensics?, in: Proceedings of the 15th International Conference on Multimedia, 2007, pp. 78–86.

[3] C. Rey, J.L. Dugelay, A survey of watermarking algorithms for image authentication, EURASIP J. Appl. Signal Process. 1 (2002) 613–621.

[4] N.M. Yeung, Digital watermarking introduction, CACM 41 (1998) 31–33.

[5] J. Fridrich, Methods for tamper detection in digital images, in: Proceedings of the ACM Workshop on Multimedia and Security, 1999, pp. 19–23.

[6] J. Fridrich, D. Soukal, J. Lukas, Detection of copy–move forgery in digital images, in: Proceedings of Digital Forensic Research Workshop, 2003, pp. 19–23.

[7] W. Luo, Z. Qu, F. Pan, J. Huang, A survey of passive technology for digital image forensics, Front. Comput. Sci. China 1 (2009) 308–322.

[8] D. Gabor, Theory of communication, J. Inst. Electric. Eng. 93 (1946) 429–457.

[9] A. Popescu, H. Farid, Exposing Digital Forgeries by Detecting Duplicated Image Regions, Technical Report TR2004-515, Department of Computer Science, Dartmouth College, 2004.

[10] W. Luo, J. Huang, G. Qiu, Robust detection of region duplication forgery in digital image, in: Proceedings of the 18th International Conference on Pattern Recognition, vol. 4, 2006, pp. 746–749.

[11] X. Kang, S. Wei, Identifying tampered regions using singular value decomposition in digital image forensics, in: Proceedings of International Conference on Computer Science and Software Engineering, 2008, pp. 926–930.

[12] S. Bayram, H.T. Husrev, N. Memon, An efficient and robust method for detecting copy-move forgery, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2009, pp. 1053–1056.
[13] B. Mahdian, S. Saic, Detection of copy-move forgery using a method based on blur moment invariants, Forensic Sci. Int. 171 (2007) 180–189.
[14] L. Li, S. Li, H. Zhu, An efficient scheme for detecting copy-move forged images by local binary patterns, J. Inf. Hiding Multimedia Signal Process. 4 (2013) 46–56.
[15] G. Lynch, F.Y. Shih, H.M. Liao, An efficient expanding block algorithm for image copy-move forgery detection, Inf. Sci. 239 (2013) 253–265.
[16] J. Zhao, J. Guo, Passive forensics for copy-move image forgery using a method based on DCT and SVD, Forensic Sci. Int. 233 (2013) 158–166.
[17] H. Lin, C. Wang, Y. Kao, Fast copy-move forgery detection, WSEAS Trans. Signal Process. 5 (5) (2009) 188–197.
[18] S. Ryu, M. Lee, H. Lee, Detection of copy-rotate-move forgery using Zernike moments, in: Information Hiding Conference, June 2010, pp. 51–65.
[19] I-Cheng Chang, J. Cloud Yu, Chih-Chuan Chang, A forgery detection algorithm for exemplar-based inpainting images using multi-region relation, Image Vis. Comput. 31 (2013).
[20] Zaoshan Liang, Gaobo Yang, Xiangling Ding, Leida. Li, An efficient forgery detection algorithm for object removal by exemplar-based image inpainting, J. Vis. Commun. Image Representation 30 (2015) 75–85.
[21] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, G. Serra, A SIFT-based forensic method for copy-move attack detection and transformation recovery, IEEE Trans. Inf. Forensics Secur. 6 (2011) 1099–1110.
[22] X. Pan, S. Lyu, Region duplication detection using image feature matching, IEEE Trans. Inf. Forensics Secur. 4 (2010) 857–867.
[23] Alexandra Gilinsky, Lihi Zelnik Manor, SIFTpack: a compact representation for efficient SIFT matching, in: IEEE International Conference on Computer Vision, 2013, pp. 777–784.
[24] B.L. Shivakumar, S. Baboo, Detection of region duplication forgery in digital images using SURF, Int. J. Comput. Sci. 8 (2011) 199–205.
[25] L. Chen, W. Lu, J. Ni, W. Sun, J. Huang, Region duplication detection based on Harris corner points and step sector statistics, J. Vis. Commun. Image Representation 24 (2013) 244–254.
[26] E. Silva, T. Carvalho, A. Ferreira, A. Rocha, Going deeper into copy-move forgery detection: exploring image telltales via multi-scale analysis and voting processes, J. Vis. Commun. Image Representation 29 (2015) 16–32.
[27] Wei-Ying Ma, B.S. Manjunath, A texture thesaurus for browsing large aerial photographs, J. Am. Soc. Inf. Sci. 49 (7) (1998) 633–648. Wiley for ASIS.
[28] W.Y. Ma, B.S. Manjunath, Texture features and learning similarity, in: Proc. IEEE International Conference on Computer Vision and Pattern Recognition, San Francisco, CA, June 1996, pp. 425–430.
[29] J. Daugman, Two-dimensional analysis of cortical receptive field profiles, Vision. Res. 20 (1980) 846–856.
[30] J. Daugman, Uncertainty relation for resolution in space, spatial frequency and orientation optimized by two-dimensional visual cortical filters, J. Opt. Soc. Am. 2 (1985) 1160–1169.
[31] S. Marcelja, Mathematical description of the responses of simple cortical cells, J. Opt. Soc. Am. 70 (1980) 1297–1300.
[32] D. Zhang, W.K. Kong, J. You, M. Wong, Online palmprint identification, IEEE Trans. Pattern Anal. Mach. Intell. 25 (2003) 1041–1050.
[33] CoMoFoD database. <http://www.vcl.fer.hr/comofod>.
[34] Image Manipulation Dataset. <http://www5.cs.fau.de/research/data/image-manipulation>.
[35] A. Criminisi, P. Perez, K. Toyama, Object removal by exemplar-based inpainting, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, 2003, pp. II-721–II-728.
[36] J.H. Choi, C.H. Hahm, An exemplar-based image inpainting method with search region prior, in: 2013 IEEE 2nd Global Conference on Consumer Electronics, 2013, pp. 68–71.
[37] Vincent Christlein, Christian Riess, Johannes Jordan, Corinna Riess, Elli Angelopoulou, An evaluation of popular copy-move forgery detection approaches, IEEE Trans. Inf. Forensics Secur. (TIFS) 7 (2012) 1841–1854.