Multimed Tools Appl (2018) 77:363–385
DOI 10.1007/s11042-016-4276-3

Multi-scale feature extraction and adaptive matching for copy-move forgery detection

XiuLi Bi[1] · **Chi-Man Pun**[1] · **Xiao-Chen Yuan**[2]

Received: 12 April 2016 / Revised: 30 October 2016 / Accepted: 15 December 2016 /
Published online: 26 December 2016

Abstract A copy-move forgery detection scheme by using multi-scale feature extraction and adaptive matching is proposed in this paper. First, the host image is segmented into the non-overlapping patches of irregular shape in different scales. Then, Scale Invariant Feature Transform is applied to extract feature points from all patches, to generate the multi-scale features. An Adaptive Patch Matching algorithm is subsequently proposed for finding the matching that indicate the suspicious forged regions in each scale. Finally, the suspicious regions in all scales are merged to generate the detected forgery regions in the proposed Matched Keypoints Merging algorithm. Experimental results show that the proposed scheme performs much better than the existing state-of-the-art copy-move forgery detection algorithms, even under various challenging conditions, including the geometric transforms, such as scaling and rotation, and the common signal processing, such as JPEG compression and noise addition; in addition, the special cases such as the multiple copies and the down-sampling are also evaluated, the results indicate the very good performance of the proposed scheme.

Keywords Copy-Move Forgery Detection · Multi-Scale Feature Extraction · Adaptive Patch Matching

✉ Chi-Man Pun
cmpun@umac.mo

XiuLi Bi
yb47429@umac.mo

Xiao-Chen Yuan
xcyuan@must.edu.mo

[1] Department of Computer and Information Science, University of Macau, Macau, China

[2] Faculty of Information Technology, Macau University of Science and Technology, Macau, China

1 Introduction

With the fast development of computer technology and the popularity of software for image processing, image forgery which greatly reduces the credibility of the digital images, is becoming much easier to be achieved. Therefore, the image forgery detection has been becoming more and more attractive in recent years. The copy-move forgery is to paste a region / regions of an image into another part(s) of the same image. During the copy and move operation, some image processing methods such as rotation, scaling, blurring, and compression are always applied to ensure the imperceptibility of the copied region(s); however, they will increase the difficulties of forgery detection at the same time. On the other hand, since the copy and move operations are executed in the same image, which means the noise component, color characters and other important properties of the pasted region(s) are compatible with which of the rest of the image. Therefore, some of the forgery detection methods based on the related image properties are not applicable in this case. In past years, lots of methods have been proposed for the copy-move forgery detection, of which two main categories of features are usually employed: the block based features and the keypoint based features.

The block based algorithms [5–7, 11, 12, 14, 15, 17, 19, 20, 22, 24, 27, 28] are usually to divide the host images into blocks, extract the block features, and find the tampered regions from the matched blocks which have similar block features. Lots of block features have been proposed and employed in the area of forgery detection. Popescu and Farid [22] applied the Principal Component Analysis (PCA) method to reduce the feature dimensions. Luo *et al.* [19] used the RGB color components and direction information as block features. Li *et al.* [15] used Discrete Wavelet Transform (DWT) and Singular Values Decomposition (SVD) to extract block features. Mahdian and Saic [20] calculated the 24 Blur-invariant moments as block features. Kang and Wei [14] calculated the singular values of a reduced-rank approximation in each block. Bayram *et al.* [5] used the Fourier-Mellin Transform (FMT) to obtain features. Wang *et al.* used the mean intensities of circles with different radii around the block center as block features in [27] and [28]. Lin *et al.* [17] used the gray average results of each block and its sub blocks as the block features. Ryu *et al.* [24] used Zernike moments as block features. Fridrich *et al.* [12] calculated the Discrete Cosine Transform (DCT) coefficients as block feature. Bravo-Solorio and Nandi [7] calculated the information entropy as block features. Bi *et al.* [6] used the Polar Complex Exponential Transform moments as block features. Although the block based algorithms are effective in forgery detection, they have two main drawbacks: 1) the host images are usually divided into overlapping blocks of regular shape, therefore the computational complexity of block matching will accordingly increase with image size; 2) most of the existing block based algorithms cannot deal with significant geometrical transforms very well.

On the other hand, the keypoint based algorithms [2, 3, 8, 13, 21, 25, 26, 29–31] are to extract an appropriate selection of keypoint features that can guarantee the robustness against geometrical transforms, and to match the keypoints to each other, to locate the tampered regions. In [2, 8–10, 13, 21], the Scale Invariant Feature Transform (SIFT) [18] was used to extract keypoint feature. In [26, 29], the Speeded Up Robust Features (SURF) [4] was used instead of SIFT as keypoint features. In [25] DAISY have also been considered for feature extraction for CMFD methods. Since the number of keypoints is much less than the number of overlapping blocks, the computational complexity is comparatively less. However, most of methods in this category could not achieve high *precision* rate while sustain high *recall* rate [9]. Recently, several new copy-move forgery detection schemes that employ both block

features and keypoint features have been proposed. Li *et al.* [16] first segmented the image into semantically independent patches/blocks prior to keypoints extraction, then the matching between the patches/blocks are found to locate the copy-move regions. In this case, the keypoints extracted with SIFT can be regarded as block features since being extracted from patches/blocks. Similarly, we proposed an adaptive over-segmentation and feature points matching (ASFPM) method [23] for image forgery detection, which integrates the characteristics of both block features and keypoint features. The ASFPM method was proved to be superior to many state-of-the-art existing methods. In ASFPM, Discrete Wavelet Transform (DWT) was employed to analyze the frequency distribution of the host image, to adaptively determine the initial size of superpixel, however, when the host image consists of complex and mixed content, for example, smoothed textures are mixed with detailed textures, the frequency distribution calculated with DWT cannot determine the best initial size of superpixel, which will cause bad segmentation and bring inaccurate detection results. In addition, when the host image contains copied regions of different sizes, the ASFPM method will probably no longer work well.

In this paper, considering the weakness of the ASFPM method [23] as we mentioned, we propose a novel multi-scale feature extraction and adaptive matching method to detect the image copy-move forgery. In the proposed scheme, we integrate the characteristics of both block features and keypoint features to achieve better detection results, like [16, 23]. First, we segment the host image into non-overlapping patches of irregular shape in different scales; then, we apply SIFT to extract feature points from all patches, to generate the multi-scale features. An Adaptive Patch Matching algorithm is subsequently proposed for finding the matching that can indicate the suspicious regions in each scale. Finally, the suspicious regions in all scales are merged to determine the detected forgery regions. In the next section 2, we will give the framework of the proposed copy-move forgery detection scheme and explain each step in detail. In section 3, a lot of experiments are conducted to demonstrate the effectiveness of the proposed scheme. Finally, the conclusions are drawn in section 4.

2 The proposed copy-move forgery detection scheme

The proposed scheme using multi-scale feature extraction and matching integrates the characteristics of both block features and keypoint features and performs very well when there are multiple copy-move objects/regions and especially when the objects/regions are of different sizes and contain both smoothed and detailed textures. Figure 1 demonstrates the advantage of the proposed multi-scale based method, compared with the ASFPM [23] which we have proposed in our previous work. The compared results are estimated by *precision* and *recall* rate: *precision* is defined as the ratio of number of correctly detected forged pixels to the number of totally detected forged pixels; *recall* is defined as the ratio of number of correctly detected forged pixels to the number of forged pixels in the ground-truth forged image. In the first row of Fig. 1, (a) shows the forgery image (image size:3888 × 2592), where three objects of different sizes and different textures are copied and pasted; (b) shows its corresponding ground truth image; (c) shows the detected results of the ASFPM method [23], from which the detection accuracy is calculated as: *precision=84.65%*, *recall=70.54%*; and (d) shows the detected results of the proposed multi-scale based scheme, from which the detection accuracy is calculated as: *precision=97.96%*, *recall=85.72%*. In this case, it is obvious and perceptual that the proposed scheme performs much better than the ASFPM method [23]. Similarly, in the

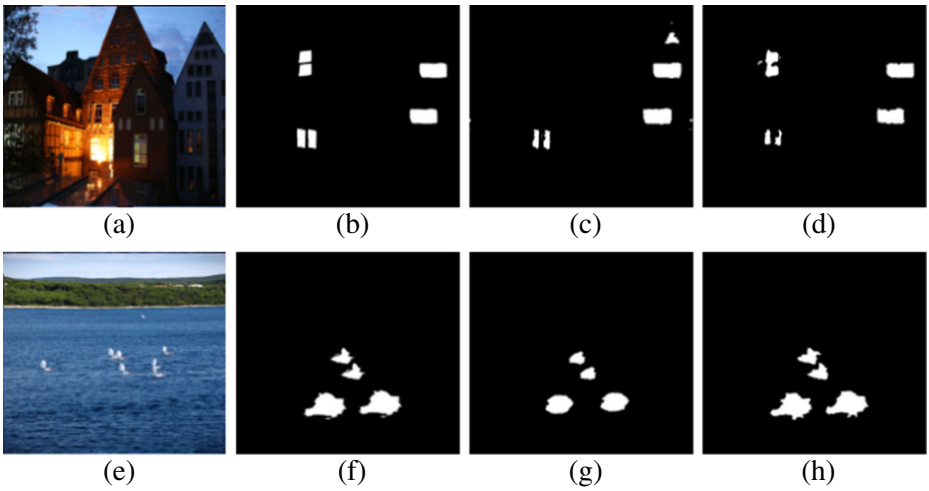

Fig. 1 Demonstration of detection results of ASFPM method and proposed method. 1st column: the forgery images; 2nd column: ground truth images; 3rd column: the detected regions of ASFPM method; and 4th column: the detected regions of the proposed scheme

second row of Fig. 1, another example (image size:2613 × 3900) is demonstrated. Two objects of different sizes are copied and pasted in (e), (e) and (f) respectively show the forgery image and ground truth image; (g) shows the detected results of the ASFPM method [23], with detection accuracy as: *precision=96.90%*, *recall=78.90%*; and (h) shows the detected results of the proposed scheme, with detection accuracy as: *precision=96.21%*, *recall=98.67%*. In this case, the *precision* rates of the two methods are similar, however the proposed scheme outperforms the ASFPM method [23] a lot in respect of *recall* rate.

Figure 2 shows the framework of the proposed scheme. First, the Multi-Scale Feature Extraction (MSFE) algorithm is proposed and applied to the host image, to generate the Multi-Scale Features (MSF). Then, the Adaptive Patch Matching (APM) algorithm is proposed and applied to the MSF, to obtain the matched patch pairs; from which the Matched Keypoints (MK) are obtained. Finally, according to MK, the Matched Keypoints Merging (MKM) algorithm generates the detected forgery regions. In the remainder of this section, sections *2.1*, *2.2* and *2.3* will explain the three proposed algorithms: MSFE, APM, and MKM, respectively, in details.

2.1 Multi-scale feature extraction algorithm

In the existing block based algorithms [5, 7, 14, 15, 17, 19, 20, 22, 24, 27, 28], the host image is divided into the overlapping regular blocks (e.g. block size =16 × 16 in Fig. 3-(a)); then, the forgery regions can be found from the matched blocks. Since, the forgery regions are not of regular shapes (e.g. in Fig. 4-(a)), the detected forgery regions represented by the set of regular blocks are usually inaccurate. In addition, when the size of host image increases, the computational time of the block matching will be much more expensive. To address the above-mentioned problems, we propose to segment the host image into non-overlapping irregular patches (e.g. in Fig. 3-(b)), and then to find the forgery regions by matching the non-overlapping and irregular patches.

Considering that superpixel algorithms can group pixels into perceptually meaningful atomic regions, in our algorithm, we employ the Simple Linear Iterative Clustering (SLIC) algorithm [1] to segment the host image. SLIC algorithm adapts a k-means clustering approach to efficiently

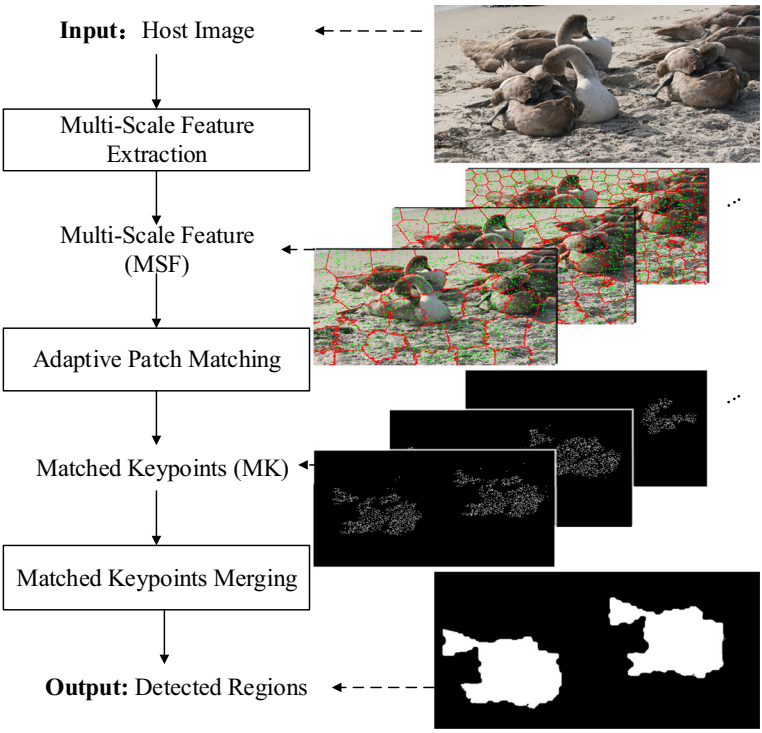

Fig. 2 The framework of the proposed copy-move forgery detection scheme

generate superpixels, and superpixel adheres to boundaries very well. With SLIC, the host image is segmented into the non-overlapping superpixels that are meaningful and are of irregular shapes. The non-overlapping segmentation method can help to decrease the computational expenses, compared with the existing overlapping block method; in addition, in most of the cases, the irregular and meaningful regions can represent the forgery regions better than the regular blocks. Figure 3 shows the different blocking/segmentation methods, where (a) shows the overlapping and regular blocking method of the existing forgery detection algorithms and (b) shows the non-overlapping and irregular segmentation method of the proposed scheme.

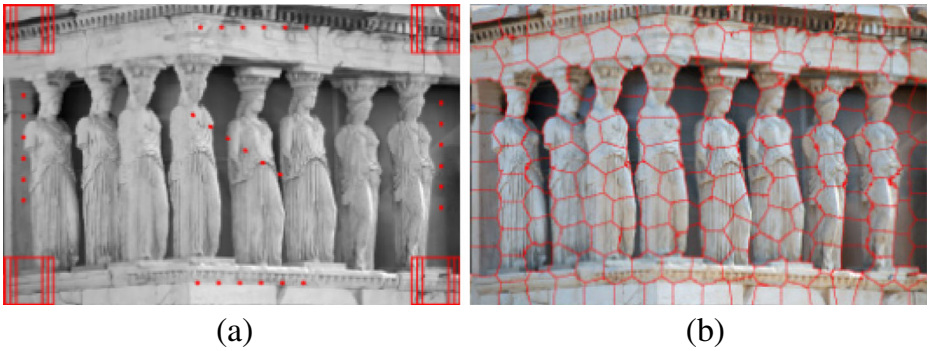

Fig. 3 Different blocking/segmentation methods. **a** Overlapping and regular blocking; (**b**) Non-overlapping and irregular segmentation

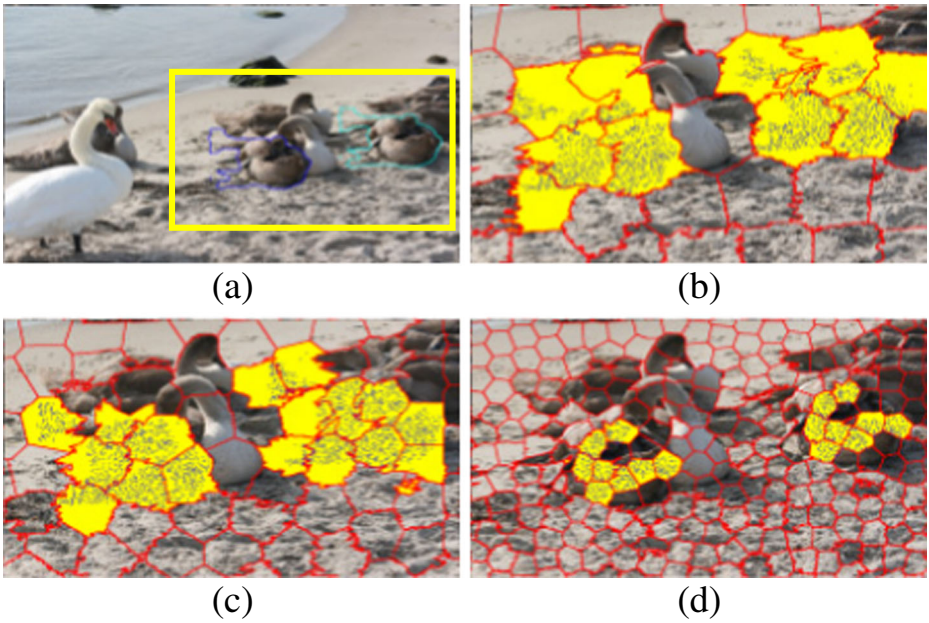

Fig. 4 Three-scale segmentation demonstration. **a** The forgery image; (**b**, **c**) and (**d**) The segmentation results in three different scales and the corresponding matched patches and the matched keypoints selected from the matched patches

Most of the existing block based algorithms divided the host images only in single scale with initially predefined block size; in that situation, if block size is too small, some forgery regions will be missed, as shown in Fig. 4-(d); otherwise, if the block size is too large, some error detected results will be introduced, as shown in Fig. 4-(b). To solve this problem, we propose the multi-scale segmentation in our algorithm. Figure 4 shows the three-scale segmentation and the corresponding segmentation result in each scale. In Fig. 4, (a) shows the forgery image, where the region in sky-blue represents the original region, and the region in blue represents the target copy-move region; (b), (c) and (d) respectively show the segmentation results in three different scales. The patches highlighted in yellow indicate the matched patches, while the points highlighted in blue indicate the matched keypoints.

As discussed above, the proposed MSFE algorithm segments the host image into the patches with multiple scales; from each patch the feature points are then extracted. In recent years, the feature points extraction methods such as SIFT [18] and SURF [4] are widely used in the field of computer vision. SIFT and SURF were proved to be robust against the common image processing operations such as rotation, scale, blurring, and compression; in consequence, SIFT and SURF were also used as feature points extraction methods in most of the existing keypoint based forgery detection algorithms. Christlein et al. [9] compared performances of SIFT and SURF with the other 13 image feature extraction methods in the comparative experiments, and the results indicate that the SIFT and SURF based methods perform better than the others. Therefore, in this scheme, we choose SIFT with default parameters as the feature extraction method to extract feature points as patch features.

Figure 5 shows the flowchart of the proposed MSFE algorithm. First, the host image is blocked into the patches with the superpixel segmentation method; then, the feature points are extracted from these patches. The whole process is repeated along with the decreasing of the

size of segmentation, until feature points cannot be extracted any more in the corresponding scale. Finally, the multi-scale feature *MSF* is generated, which includes the patches in each scale and the corresponding feature points. The steps of the proposed MSFE algorithm are explained in Algorithm I as follows.

Algorithm I: Multi-Scale Feature Extraction (MSFE) Algorithm

Input: Host image;

Output: Multi-Scale Feature MSF.

STEP-1: Load the host image and initialize the initial scale $n=1$, the initial number of blocks $B_n = B$, the initial set of patch feature $PF^n = \varnothing$, and the initial set of multi-scale feature $MSF = \varnothing$.

STEP-2: Apply the SLIC algorithm to segment the input image into B_n patches P^n, $P^n = \left\{P_1^n, P_2^n, P_3^n, \ldots, P_{B_n}^n\right\}$.

STEP-3: Apply SIFT algorithm to each patch to extract feature points F^n, $F^n = \left\{F_1^n, F_2^n, F_3^n, \ldots, F_{B_n}^n\right\}$.

STEP-4: Organize the set of patch feature $PF^n = \left\{P^n, F^n\right\}$; and the set of multi-scale feature MSF as $MSF = MSF \bigcup PF^n$.

STEP-5: Check the existence of the extracted feature points F^n, if $F^n \neq \varnothing$, $n = n+1$, $B_n = 4^{n-1} * B_{n-1}$, repeat STEP-2 to STEP-4; otherwise, output the set of multi-scale feature MSF, $MSF = \left\{PF^1, PF^2, \ldots, PF^n\right\}$.

In STEP-1 of Algorithm I, the appreciate initialization of *B* can avoid segmenting the host image into excessive scales. In the experiments, by experiments, the *B* is initially set as 200 when the size of host image $M \times N$ is larger than1500×1500; otherwise, the *B* is initially set as 100.

2.2 Adaptive patch matching algorithm

After generating*MSF*, we need to locate the matched patch pairs in each scale. In most of the existing block based algorithms, the block matching generates specific block pairs only if there are many other matched pairs in the same mutual position, assuming they have the same shift vector. When the number of matched block pairs, which have same shift vector, exceeds a user-specified threshold, the matched block pairs that contribute to that specific shift vector will be identified as the regions that probably have been tampered. In that situation, the

threshold is related to the regions that can be identified; the larger threshold may cause some not-so-closely matched blocks missing, while the smaller threshold may bring more false matched blocks. Therefore, the threshold highly relates with the performance of the forgery detection algorithms, and how to determine the just right threshold becomes an important issue.

An Adaptive Patch Matching (APM) algorithm which aims at improving the existing matching process is proposed by adaptively determining the threshold. Figure 6 shows the flowchart of the APM algorithm. In the i^{th} scale ($i \in \{1, \cdots, n\}$), the number of matched keypoints of each patch pair is calculated according to $PF^i = \{P^i, F^i\}$ and the correlation coefficient map CC^i is generated; then the corresponding patch matching threshold TP^i is determined adaptively; the matched patch pairs MP^i are located by TP^i; and finally the matched keypoints MK^i are selected from MP^i. The steps of the proposed APM algorithm are explained in Algorithm II as follows.

Algorithm II: Adaptive Patch Matching (APM) Algorithm

Input: Multi-Scale Feature ***MSF***;

Output: Matched Keypoints ***MK***.

STEP-1: Load the Multi-Scale Feature $MSF = \left\{PF^1, PF^2, \ldots, PF^n\right\}$, where n means the number of scales; $PF^n = \left\{P^n, F^n\right\}$ is the set of patch feature.

STEP-2: In each scale, calculate the numbers of matched keypoints between each two patches, which are defined as correlation coefficient of the corresponding patch pair; and thus generate the correlation coefficient map $CC = \left\{CC^1, CC^2, \ldots, CC^n\right\}$.

STEP-3: According to CC, adaptively calculate the value of patch matching threshold as $TP = \left\{TP^1, TP^2, \ldots, TP^n\right\}$.

STEP-4: According to the corresponding matching threshold TP, locate the matched patch pairs MP as $MP = \left\{MP^1, MP^2, \ldots, MP^n\right\}$.

STEP-5: Extract the matched keypoints MK in MP as $MK = \left\{MK^1, MK^2, \ldots, MK^n\right\}$.

In STEP-2 of Algorithm II, the keypoints are matched using the *best-bin-first* algorithm with their Euclidian distance; which means that a keypoint f_a is matched to the keypoint f_b only if they can meet the following condition:

$$d(f_a, f_b) \cdot TK \leq d(f_a, f_i) \tag{1}$$

Where $d(f_a, f_b)$ means the Euclidian distance between the keypoints f_a and f_b, and it is defined in (2); $d(f_a, f_i)$ means the Euclidian distances between the keypoints f_a and all other keypoints,

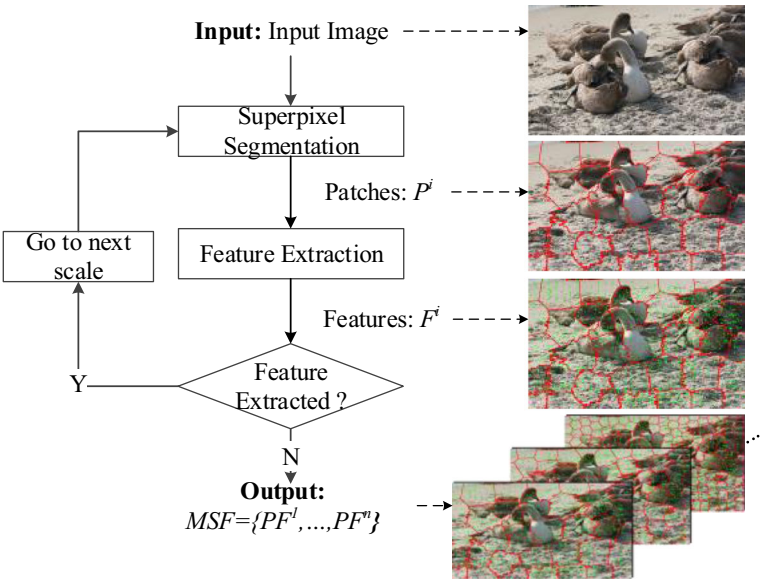

Fig. 5 Flowchart of the Multi-Scale Feature Extraction (MSFE) algorithm

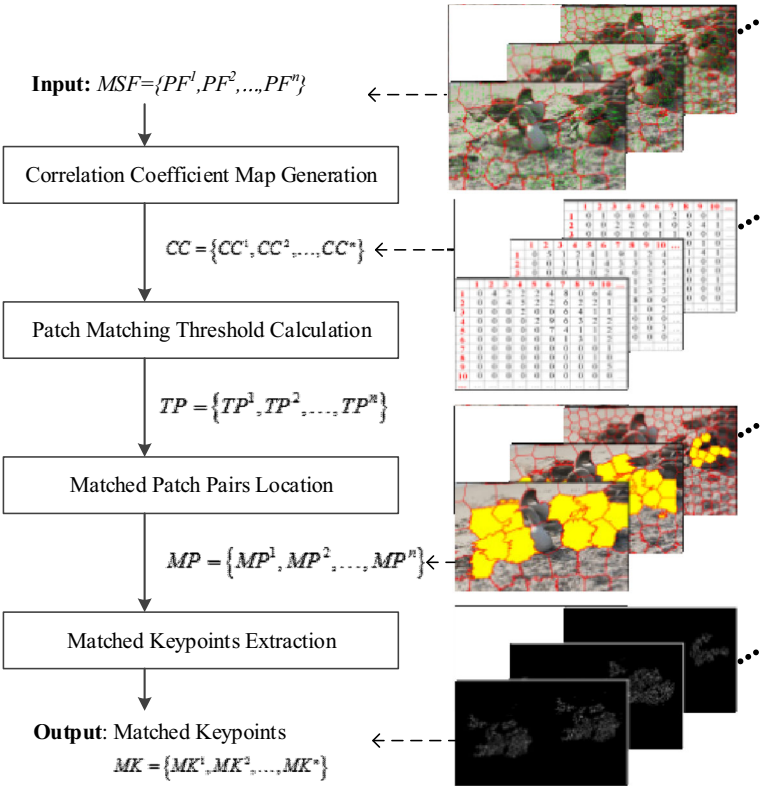

Fig. 6 Flowchart of the Adaptive Patch Matching (APM) algorithm

and it is defined in (3). *TK*is the keypoints matching threshold; when *TK* becomes larger, the matching accuracy will be higher, but meanwhile the ratio outliers will be higher accordingly, which will cause greater miss probability. Therefore, in the experiments, we set $TK = 2$ by experiments to provide a good trade-off between matching accuracy and miss probability.

$$d(f_a, f_b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \tag{2}$$

$$d(f_a, f_i) = \sqrt{(x_a - x_i)^2 + (y_a - y_i)^2}, i = 1, 2, ...n; i \neq a, i \neq b \tag{3}$$

Correlation coefficient means the number of matched keypoints between the two patches. Assuming there are B_i patches in the i^{th} scale, we can generate $t = B_i(B_i - 1)/2$ correlation coefficients, which form the correlation coefficient mapCC^i. After generating$CC = \{CC^1, CC^2, \ldots, CC^n\}$, we need to calculate the patches matching threshold *TP* as stated in STEP-3 of Algorithm II. The procedures of the adaptive calculation of the patch matching threshold *TP* in each scale are explained as follows in Algorithm III.

Algorithm III: Adaptive Patch Matching Threshold Calculation

STEP-1: Sort the correlation coefficients in ascending order as $CC\_S^i = \left\{CC_1^i, CC_2^i, CC_3^i, \cdots\cdots, CC_t^i\right\}$, where i means in the i^{th} scale, and t means the number of correlation coefficients in the corresponding scale, $t = B_i(B_i - 1)/2$; and filter out the repeated correlation coefficients as $CC\_F^i = \left\{CC_1^i, CC_2^i, CC_3^i, \cdots\cdots, CC_f^i\right\}$, where $f \leq B_i(B_i - 1)/2$.

STEP-2: Calculate the first derivative of CC_F^i, $\nabla(CC\_F^i)$; the mean value of the first derivative vector, $\overline{\nabla(CC\_F^i)}$; and the second derivative of CC_F^i, $\nabla^2(CC\_F^i)$.

STEP-3: Select the correlation coefficients $CC\_F_j^i$, of which their second derivative is larger than the mean value of the corresponding first derivative vector, as defined in (4).

$$\nabla^2(CC\_F_j^i) > \overline{\nabla(CC\_F^i)} \tag{4}$$

STEP-4: Extract the minimum value from $CC\_F_j^i$ and set its correlation coefficient value as the corresponding patch matching threshold TP^i.

$$\nabla^2\left(CC\_F_j^i\right) > \overline{\nabla(CC\_F^i)} \tag{4}$$

Figure 7 shows the demonstration of patch matching threshold calculation. In Fig. 7, (a1) shows the forgery image; (a2) shows the plot of the sorted and filtered correlation coefficients

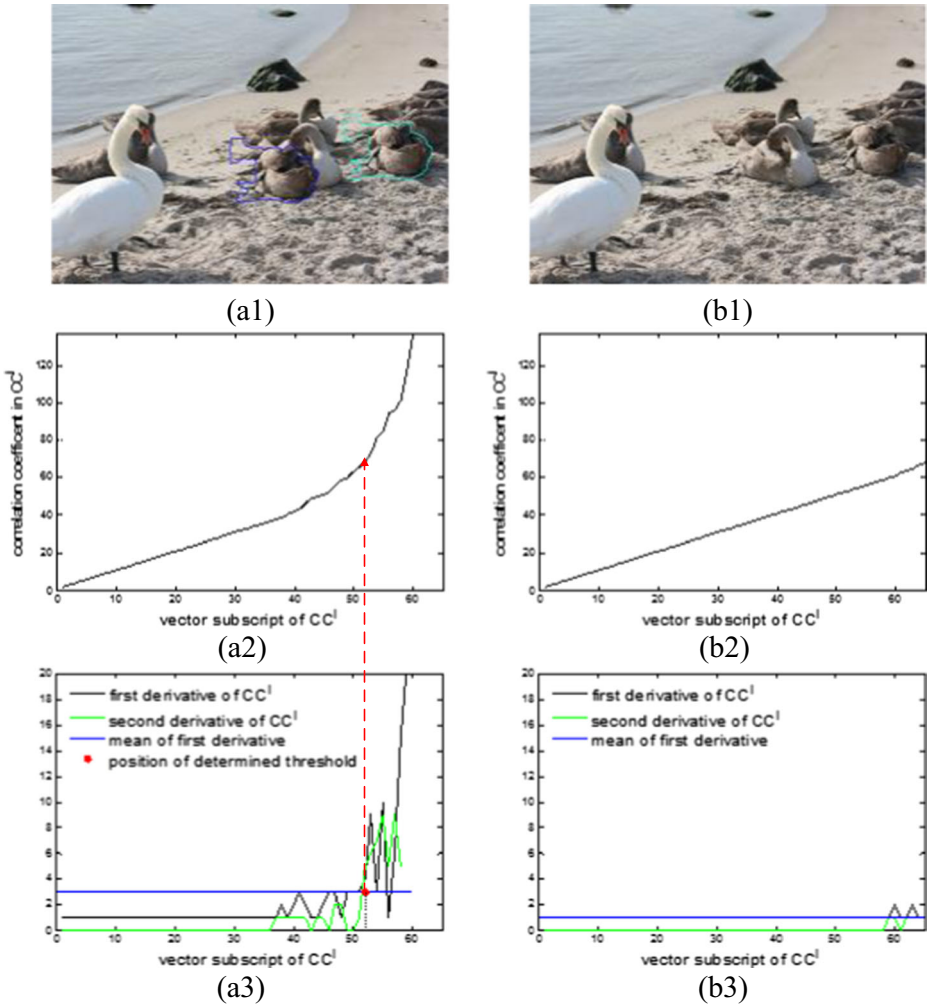

Fig. 7 Demonstration of patch matching threshold calculation. (a1) Forgery image; (a2) Plot of the correlation coefficients after sorting and filtering, of the forgery image, in the i^{th} scale; (a3) Plots of the first derivative, the second derivative, and the mean of first derivative, calculated from (a2); (b1) Original image; (b2) Plot of the correlation coefficients after sorting and filtering, of the original image, in the i^{th} scale; (b3) Plots of the first derivative, the second derivative, and the mean of first derivative, calculated from (b2)

of the i^{th} scale of forgery image, which are calculated with STEP-1 of Algorithm III; and (a3) shows the plots of the first derivative, the second derivative, and the mean of first derivative of the i^{th} scale of forgery image, which are calculated from (a2) with STEP-2 of Algorithm III. According to STEP-3 and STEP-4 of Algorithm III, the corresponding threshold is calculated and located in (a3), represented by the red point. Meanwhile, (b1) shows the original image without forgery; (b2) shows the corresponding plot of the sorted and filtered correlation coefficients of the i^{th} scale; and (b3) shows the corresponding plots of the first derivative, the second derivative, and the mean of first derivative of the i^{th} scale, which are calculated from (b2). It can be easily seen that we cannot get the

matching threshold from the non-forgery image, when using the proposed Adaptive Matching Threshold Calculation algorithm. After calculating the patch matching threshold of each scale adaptively, we can locate the matched patch pairs in each scale if their correlation coefficients are larger than the corresponding matching threshold. From those matched patches in each scale, we selected the matched keypoints to form the Matched Keypoints (*MK*).

2.3 Matched keypoints merging algorithm

After obtaining the matched keypoints *MK*, we need to determine the forgery regions by turning the independent pixels/keypoints into regions. Figure 8 shows the flowchart of the MKM algorithm. First, the host image is segmented into small superpixels; then, *MK* are replaced by the small superpixels to form the suspected forgery regions. The size of small superpixels is related with the size of the host image; when the host image is of higher resolution, the size of small superpixels will be larger. In our test dataset, the average size is approximate3000 × 2000, therefore, we set the size of small superpixel as 20 by experiments. Next, the suspected forgery regions in all scales are merged. If the suspected forgery regions are merged together by using 'OR' operation, the miss rate of the forgery detection will be reduced, however, the probability of error detection will be bigger, Fig. 9-(c) and (d) demonstrate the results of this case. Therefore, we need to filter out some regions which may be wrongly detected during the merging process. In all scales, we count the pixel appearing times

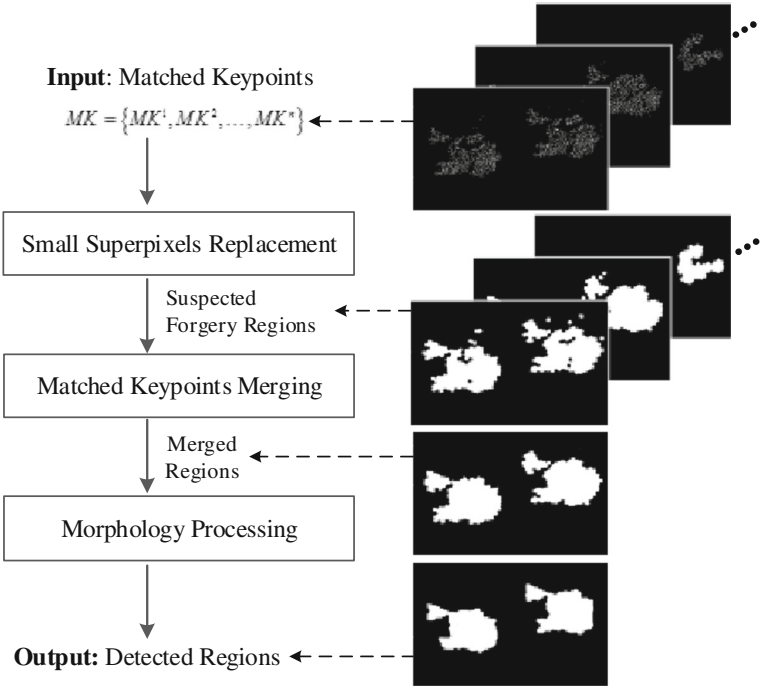

Fig. 8 Flowchart of the Matched Keypoints Merging (MKM) algorithm

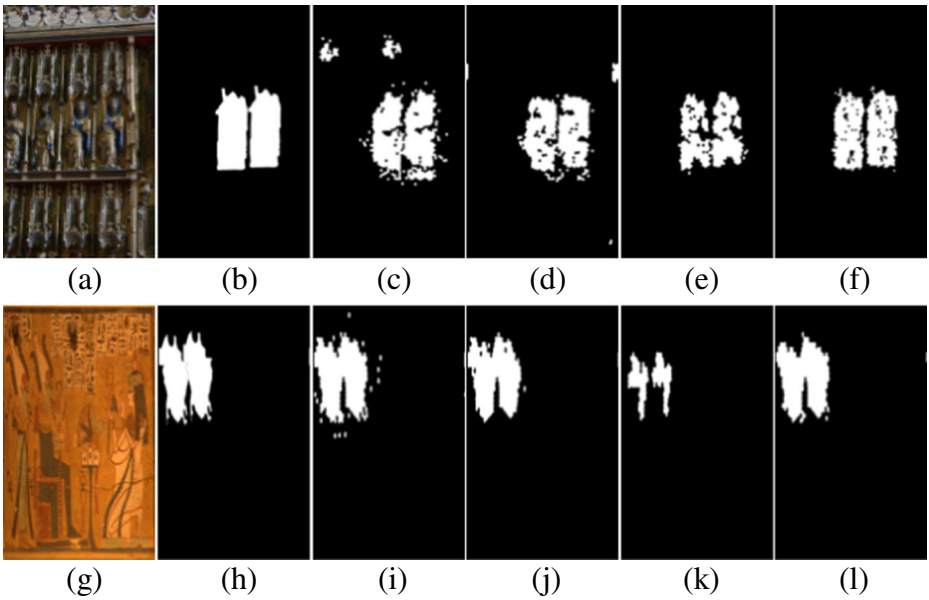

Fig. 9 The demonstration of the suspected forgery regions merging results. **a**, **g** The forgery images; (**b**, **h**) The copy-move forgery regions; (**c**, **i**), (**d**, **j**), (**e**, **k**) The suspected forgery regions in three different scales; and (**f**, **l**) The merged regions

as $T = \{t_{\min}, t_{\min+1}, \ldots, t_{\max}\}$, where $t_{\max}$ is the maximum value of the pixel appearing times in all scales, $t_{\max} \le n$. Because the host images are different, the T is considered as a random sequence, and the probability of random variable is regarded as the normal distribution. Consequently, the mean μ and the standard deviationσ of random sequence Tcan be respectively calculated by using (5) and (6).

$$\mu = \frac{1}{\max - \min} \sum_{i=\min}^{\max} t_i \tag{5}$$

$$\sigma = \sqrt{\frac{1}{\max - \min} \sum_{i=\min}^{\max} (t_i - \mu)^2} \tag{6}$$

Considering the property of normal distribution that the area of $\mu - 2\sigma$ range in the normal distribution curve is 95%, we choose $\mu - 2\sigma$ as the merging threshold to filter out the wrongly detected pixels. When the appearing time of the pixel is smaller than the merging threshold $\mu - 2\sigma$, the corresponding pixels will be discarded. Therefore, the suspected forgery regions in all scales are merged using (7), and at the same time, some of the wrongly detected regions can be discarded. Figure 9 shows the demonstration of merging results of the suspected forgery regions.

In Fig. 9, (c)(i), (d)(j) and (e)(k) display the suspected forgery regions in three scales, which indicate that the proposed multi-scale method will bring some wrong regions; while (f)(l)

display the detected regions after the merging process, where the inaccurately detected regions have been successfully removed.

$$g(x,y) = \begin{cases} 1 & \mu - 2\sigma \le \sum_{i=1}^{n} f_i(x,y) \le t_{\max} \\ 0 & 0 \le \sum_{i=1}^{n} f_i(x,y) < \mu - 2\sigma \end{cases} \tag{7}$$

Where $g(x, y)$ are the merged regions; $f_i(x, y)$ represents the suspected forgery regions in the i^{th} scale; n is the number of scales; and $\mu - 2\sigma$ is the threshold that is used to filter out the inaccurately detected pixels.

In the last step of MKM algorithm, we apply the close morphology operation to generate the final regions. The structural element we use in the close operation is defined as a circle whose radius is related to the host image size. The close operation fills the gap in the merged regions, while keeps the shape of the regions.

3 Experiments and discussions

In this section, the experiments are conducted to evaluate the effectiveness and robustness of the proposed copy-move forgery detection scheme. In the following experiments, the benchmark database[9] which consists the realistic copy-move forgeries is used to test the proposed scheme. Figure 10-(a1) ~(e1) shows a selection of images from the database. The dataset comprises 48

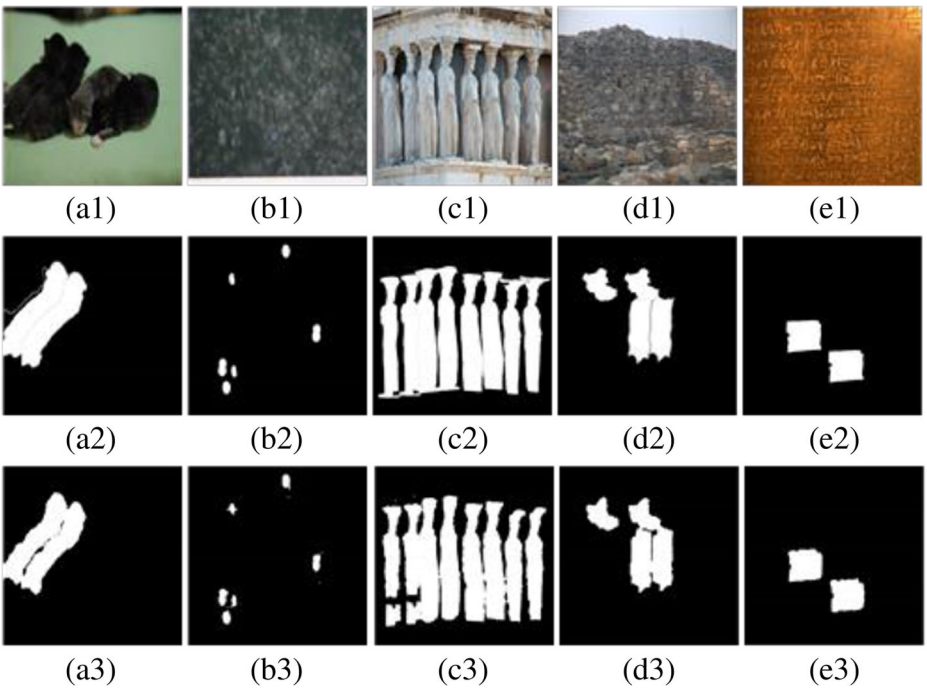

Fig. 10 The copy-move forgery detection results of the proposed scheme. The 1st row: the five selected images in the dataset; 2nd row: ground truth images; 3rd row: The detected forged regions

uncompressed PNG true color images. The average size of forgery regions is about 6% of each image. These images have a size of 3000×2300 pixels. The copied regions are of categories of living, nature, man-made and even mixed, and they range from smooth to highly texture; the copy-move forgeries are created by copying, scaling and rotating semantically meaningful image regions. JPEG compression and down-sampling are also added on the forgery images; in addition, the combined transformations and multiple copies forgeries are included in the image dataset. Therefore, we choose this dataset to objectively evaluate our scheme. Figure 10 shows the copy-move forgery detection results of the proposed scheme. In Fig. 10, the figures in the first row are the forgery images selected from the dataset; the second row displays the ground truth; and the third row displays the detected forgery regions.

In order to evaluate the performance of the proposed scheme, the two characteristics *precision* and *recall* [9] are calculated using (8) and (9) respectively. We also give the *F* score [9], which is defined in (10), as a measure which combines the *precision* and *recall* in a single value.

$$precision = \frac{\left|\Omega \cap \Omega^{'}\right|}{|\Omega|} \tag{8}$$

$$recall = \frac{\left|\Omega \cap \Omega^{'}\right|}{\left|\Omega^{'}\right|} \tag{9}$$

Where Ω means the set of forgery regions detected by the proposed scheme for the dataset; and $\Omega^{'}$ means the set of all forgery regions for the dataset.

$$F = 2 \times \frac{precision \times recall}{precison + recall} \tag{10}$$

To reduce the effect of random samples, the average *precision*/*recall* is computed over all the images in the dataset. Since Christlein *et al.* [9] have particularly recommended all benchmark methods, we use the dataset they provided and compare our experimental results with several state-of-the-art algorithms: the SIFT based detection method [9], which combined the methods of [2, 21]; the SURF based detection method [9]; Zernike moments based forgery detection method [24]; the method proposed by Bravo [8]; the SBFD method proposed in [16]; and the ASFPM method [23] which we have proposed in our previous work. We mainly compare the performances of our scheme with the state-of-the-art algorithms under different scenarios: the plain copy-move forgery; the forgery with distortion by various attacks including: scaling, rotation, Gaussian noise addition JPEG compression, and even combined attacks; the multiple copies forgery and the down-sampling forgery. The following sections *3.1*, *3.2*, and *3.3* demonstrate the detection results.

3.1 Detection results under plain copy-move forgery

Basically, we firstly evaluate the proposed scheme when under the ideal condition, that is the plain copy-move forgery. We have 48 original images and 48 forgery images, where one to one copy-move forgery is implemented. The detection methods distinguish the original images from the forgery images in this case. We evaluate the scheme at both pixel level and image level, and Tables 1 and 2 show the detection results of the 96 images at the image level and the pixel level,

Table 1 Detection results of the plain copy-move forgery at the image level

Image level	*precision* (%)	*recall* (%)	*F* (%)
SIFT [9]	88.37	79.17	83.52
SURF [9]	91.49	89.58	90.52
Zernike [24]	92.31	*100.0*	96.00
Bravo [8]	87.27	*100.0*	93.20
SBFD [16]	70.16	83.33	76.18
ASFPM [23]	*96*	*100.0*	*97.96*
Proposed Scheme	90.57	*100.0*	95.05

The italic entries indicate the best results

respectively. While pixel-level metrics are useful to assess the general localization performance of algorithm when the ground-truth data is available, and at pixel level, the *precision* and *recall* rates are calculated by counting the number of pixels in the corresponding regions; the image level decisions are particular interest to the automated detection of manipulated images, and at image level, *precision* is the probability that a detected forgery is truly a forgery, while *recall* is the probability that a forgery image is detected. In general, higher *precision* as well as higher *recall* indicates the superior performance. In Tables 1 and 2, the results in bold indicate the results of the proposed scheme and the results in bold and italic indicate the best ones. It can be easily seen that our scheme can achieve 90.57% *precision* and meanwhile 100% *recall*, which performs better than the most of existing state-of-the-art methods at image level, except the Zernike moments based method [24] which can achieve *precision* up to 92.31% and *recall* up to 100%. Meanwhile, the advantage of the proposed multi-scale detection method is particularly prominent at pixel level, when comparing with the existing state-of-the-art methods, as indicated in Table 2. The proposed method achieves *precision* up to 95.22% and *recall* up to 90.6%, which is much better than the existing state-of-the-art methods. The results indicate the good accuracy of our proposed copy-move forgery detection scheme by using multi-scale feature extraction and matching.

3.2 Detection results under various attacks

Besides the one to one plain copy-move forgery, we also test our proposed scheme when the copied regions are attacked by various attacks including geometric distortions, image degradations, and even combined attacks. That means, the forgery images are generated by using each of the 48 images in the dataset, and the copied regions are attacked by attacks as follows:

1) Scaling

The copied regions are scaled with the scale factor varies from 91% to 109%, with the step as 2%, as shown in the 1st row of Fig. 11. In this case, we need to test totally 48 × 10 = 480 images.

Table 2 Detection results of the plain copy-move forgery at the pixel level

Pixel level	*precision* (%)	*recall* (%)	*F* (%)
SIFT [9]	60.80	71.48	65.71
SURF [9]	68.13	76.43	72.04
Zernike [24]	95.07	87.72	91.25
Bravo [8]	*98.81*	82.98	89.34
SBFD [16]	84.90	54.095	65.16
ASFPM [23]	89.195	83.73	86.38
Proposed Scheme	95.22	*90.6*	*92.85*

The italic entries indicate the best results

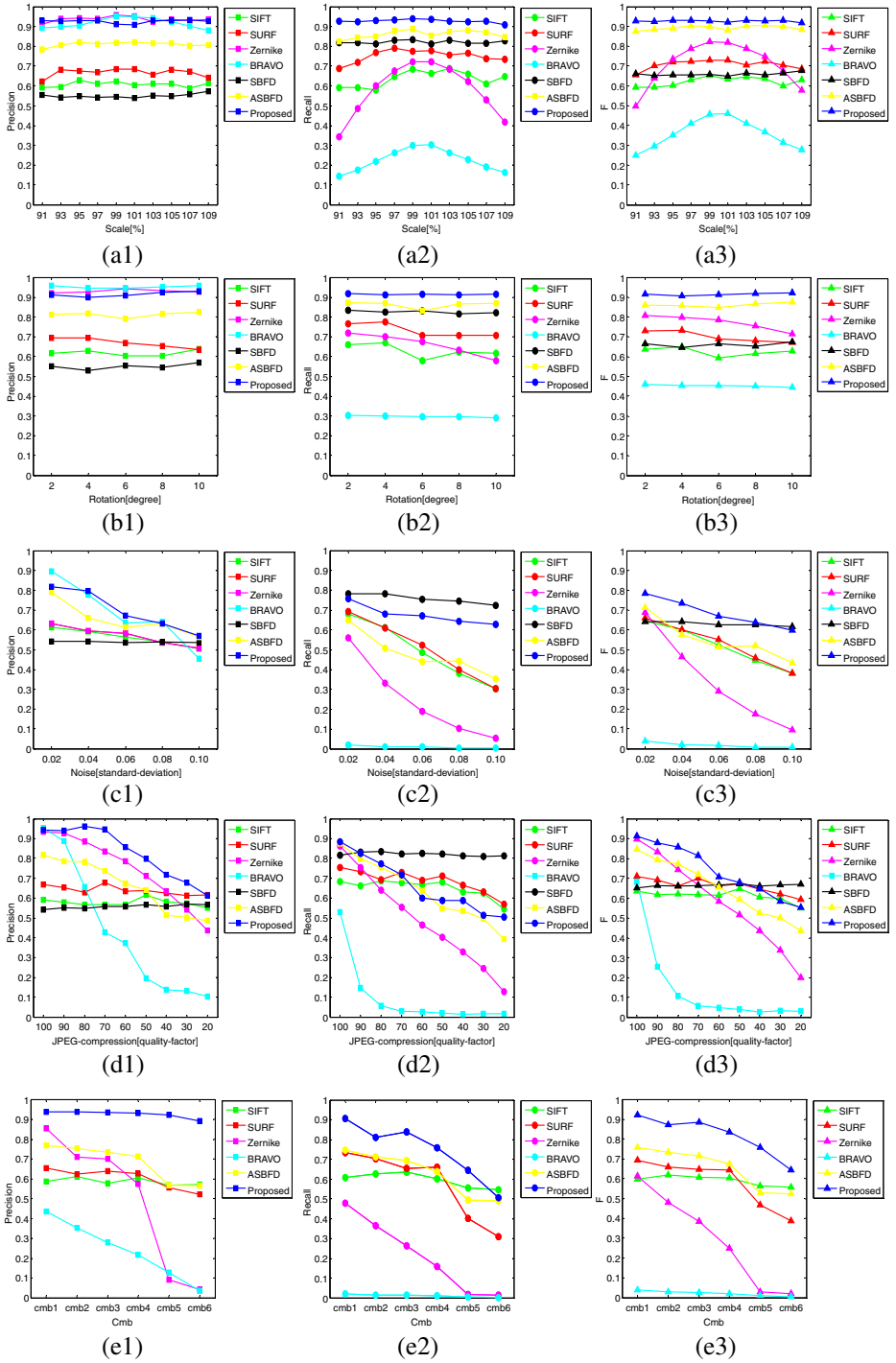

Fig. 11 Detection results under various attacks. (a1) ~(a3) Scaling; (b1) ~(b3) Rotation; (c1) ~(c3) Gaussian Noise addition; (d1) ~(d3) JPEG Compression; and (e1) ~(e3) Combined transforms. The first column displays the *precision*, the second column displays the *recall* and the third column displays the *F* score

2) Rotation

The copied regions are rotated with the rotation angle varies from 2° to 10°, in step of 2°, as shown in the 2nd row of Fig. 11. In this case, we need to test totally 48 × 5 = 240 images.

3) Gaussian Noise Addition

The image intensities are normalized between 0 and 1 and added zero-mean Gaussian noise with standard deviations of 0.02, 0.04, 0.06, 0.08 and 0.10 to the inserted snippets before splicing, as shown in the 3rd row of Fig. 11. In this case, we need to test totally 48 × 5 = 240 images.

4) JPEG compression

The JPEG compression is applied to the forgery images and original images, with the qualify factor varies from 100 to 20, with the step as -10, as shown in the 4th row of Fig. 11. In this case, we need to test totally 48 × 9 = 432 images.

5) Combined transforms

Six combined transforms are applied into the copied regions to evaluate the proposed scheme, as shown in the 5th row of Fig. 11. In Fig. 11-(e1)~(e3), 'cmb1' indicates the combined attack includes scaling with factor as 101%, rotation with angle as 2°, JPEG compression with quality factor as 80; 'cmb2' indicates the combined attack includes scaling with factor as 103%, rotation with angle as 4°, JPEG compression with quality factor as 75; 'cmb3' indicates the combined attack includes scaling with factor as 105%, rotation with angle as 6°, JPEG compression with quality factor as 70; 'cmb4' indicates the combined attack includes scaling with factor as 107%, rotation with angle as 8°, JPEG compression with quality factor as 65; 'cmb5' indicates the combined attack includes scaling with factor as 120%, rotation with angle as 20°, JPEG compression with quality factor as 60; and 'cmb6' indicates the combined attack includes scaling with factor as 140%, rotation with angle as 60°, JPEG compression with quality factor as 50. In this case, we use totally 48 × 6 = 288 images.

The detection results under various attacks are displayed in Fig. 11, where the results indicated in blue show the results of the proposed scheme and the results of three columns indicate the *precision* rate, *recall* rate and *F* score, respectively. In Fig. 11, (a1) ~(a3) show the results under the scaling, where the x-axis indicates the scale factor; (b1) ~(b3) show the results under the rotation, where the x-axis indicates the rotation angle; (c1) ~(c3) show the results under the Gaussian Noise addition, where the x-axis indicates the standard deviations; (d1) ~(d3) show the results under the JPEG compression, where the x-axis represents the quality factor; and (e1) ~(e3) show the results under the combined transforms. Furthermore, we compare the proposed scheme with the existing state-of-the-art methods: the SIFT based detection method [9], indicated in green; the SURF based detection method [9], indicated in red; the Zernike moments based forgery detection method [24], indicated in rose-red; the method proposed by Bravo [8], indicated in blue-sky; the SBFD method proposed in [16], indicated in black; and the ASFPM method [23] which we have proposed before, indicated in yellow. The results of the methods are indicated in lines of different colors as displayed in Fig. 11. It can be seen from the first and second rows, all the *recall*, *precision*, and *F* score of

Table 3 Detection results under the multiple copies forgery

Pixel level	*precision* (%)	*recall* (%)	*F* (%)
SIFT [9]	11.37	4.95	6.90
SURF [9]	37.49	21.86	27.62
Zernike [24]	83.15	22.00	34.79
Bravo [8]	*88.75*	58.27	*67.58*
ASFPM [23]	50.91	47.63	49.22
Proposed Scheme	58.2	*73.2*	64.83

The italic entries indicate the best results

the proposed scheme are greater than 90%, which indicates that the proposed scheme performs much better than the existing state-of-the-art forgery detection methods under the geometric transforms. As well, the proposed scheme performs well under the common signal processing such as Gaussian Noise addition and JPEG compression, as shown in the third and fourth rows. Note that, although our *recall* rates are worse than which of the SBFD method, the F scores are better than it under the Gaussian Noise addition and the JPEG compression. In Fig. 11-(e1) ~(e3), the proposed scheme is evaluated under six combined attacks we defined. It is obviously that the proposed scheme performs much better than the other methods.

3.3 Detection results under multiple copies and down-sampling

Besides the plain copy-move forgery and the forgeries attacked by various attacks, we also evaluate the proposed scheme when the forgery images have multiple copies. In order to test the multiple copies forgery, we have copied an 64×64 image region five times and moved them to the random locations in the image itself. Table 3 shows the comparison of the detection results in this scenario. It can be easily seen that the proposed scheme outperforms the most of existing detection methods except the method proposed by Bravo [8] which can achieve *precision* up to 88.75%, however, our scheme can achieve much higher *recall*. The results indicate the good performance of the proposed multi-scale feature extraction and the adaptive matching for copy-move forgery detection.

Considering that the performance of forgery detection algorithms usually matters with the quality of the resources, we evaluate the proposed scheme and compare it with the mentioned state-of-the-art methods under the down-sampling, as shown in Fig. 12, where (a), (b) and (c) display the *precision*, *recall* and *F* score, respectively. We scale down all the images in the plain copy-move forgery in step of 20%. Note that the parameters of detection methods are globally fixed to avoid over-fitting. In Fig. 12, the x-axis means the down-sampling factor and the results in blue indicate which of the proposed scheme while the results in other colors

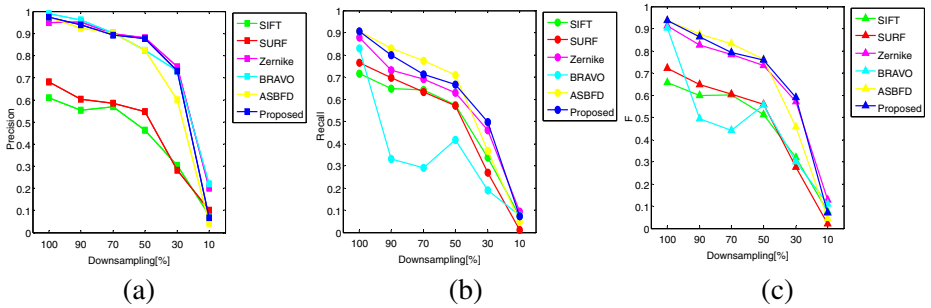

Fig. 12 Detection results under the down-sampling. **a** *precision*; (**b**) *recall*; (**c**) *F* score

indicate which of the above-mentioned state-of-the-art methods. The proposed scheme performs much better than the existing methods in this case.

4 Conclusion

With the help of the digital processing programs, images can be easily manipulated to create non-existing situations, which diminishes the credibility and value of images presented as evidence in the court or media. As one can expect, the situation will get worse, when the tools that perform the forgeries will move from research labs to commercial software. So, we must explore a method to provide a decisive answer whether an image contains image forgery. In this paper, we propose a novel multi-scale feature extraction and adaptive matching method to detect the copy-move image forgery. In the proposed scheme, first, we segment the host image by SLIC in multi-scale, to generate multi-scale patches; then we apply SIFT to patches in all the scales, to extract feature points. Next, the Adaptive Patch Matching algorithm is subsequently proposed for finding the matching which can indicate the suspicious forged regions in each scale. And finally, the suspicious regions in all scales are merged and some morphological operations are applied to generate the detected forgery regions. In general, we have four main contributions in the proposed scheme: 1) we replace the overlapping blocks of regular shape in traditional forgery detection algorithms, with individual irregular patches, which can better partition the host images into non-overlapping blocks. 2) We segment the host image into patches in multiple scales, from which the feature points are extracted respectively. The proposed multi-scale feature extraction method can extract more accurate feature points. 3) Instead of artificially setting the patch matching threshold in advance, we propose to adaptively calculate the matching threshold for better feature recognition. And 4) during the post-processing, we propose to use the predefined small superpixels to replace the matched keypoints and we apply some morphology operations into the merged regions to generate more accurately detected forgery regions.

Experimental results show that the proposed scheme performs much better than the existing state-of-the-art copy-move forgery detection algorithms, even under various challenging conditions including: the geometric transforms, such as scaling and rotation; and the common signal processing, such as JPEG compression and noise addition. In addition, the special cases such as the multiple copies and the down-sampling are also evaluated and the results indicate the very good performance of the proposed scheme. We may focus on applying multi-scale approach to other kind of forgery such as splicing or other kind of media such as video and audio in the future work.

Acknowledgement This research was supported in part by the Research Committee of the University of Macau (MYRG2015-00011-FST, MYRG2015-00012-FST) and the Science and Technology Development Fund of Macau SAR (008/2013/A1, 093-2014-A2).

References

1. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Susstrunk S (2012) SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans Pattern Anal Mach Intell 34(11):2274–2282. doi:10.1109/TPAMI.2012.120

2. Amerini I, Ballan L, Caldelli R, Del Bimbo A, Serra G (2011) A sift-based forensic method for copy–move attack detection and transformation recovery. IEEE Trans Inf Forensics Secur 6(3):1099–1110
3. Amerini I, Ballan L, Caldelli R, Del Bimbo A, Del Tongo L, Serra G (2013) Copy-move forgery detection and localization by means of robust clustering with J-linkage. Signal Process Image Commun 28(6):659–669
4. Bay H, Tuytelaars T, Van Gool L (2006) Surf: Speeded up robust features. In: Computer Vision–ECCV 2006. Springer, pp 404-417
5. Bayram S, Sencar HT, Memon N (2009) An efficient and robust method for detecting copy-move forgery. In: Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on, IEEE, pp 1053–1056
6. Bi XL, Pun CM, Yuan XC (2016) Multi-Level Dense Descriptor and Hierarchical Feature Matching for Copy–Move Forgery Detection. Inf Sci 345:226–242
7. Bravo-Solorio S, Nandi AK (2011) Exposing duplicated regions affected by reflection, rotation and scaling. In: Acoustics, Speech and Signal Processing (ICASSP), 2011 I.E. International Conference on, IEEE, pp 1880–1883
8. Bravo-Solorio S, Nandi AK (2012) Exposing postprocessed copy–paste forgeries through transform-invariant features. IEEE Trans Inf Forensics Secur 7(3):1018–1028
9. Christlein V, Riess C, Jordan J, Riess C, Angelopoulou E (2012) An evaluation of popular copy-move forgery detection approaches. IEEE Trans Inf Forensics Secur 7(6):1841–1854. doi:10.1109/Tifs.2012.2218597
10. Costanzo A, Amerini I, Caldelli R, Barni M (2014) Forensic analysis of sift keypoint removal and injection. IEEE Trans Inf Forensics Secur 9(Sept.2014):1450–1464
11. Emam M, Han Q, Niu X (2015) PCET based copy-move forgery detection in images under geometric transforms. Multimed Tools Appl:1–15. doi:10.1007/s11042-015-2872-2
12. Fridrich J, Soukal D, Lukáš J (2003) Detection of copy-move forgery in digital images. In: In Proceedings of Digital Forensic Research Workshop, Citeseer
13. Huang H, Guo W, Zhang Y (2008) Detection of copy-move forgery in digital images using SIFT algorithm. In: Computational Intelligence and Industrial Application, 2008. PACIIA'08. Pacific-Asia Workshop on, IEEE, pp 272–276
14. Kang X, Wei S (2008) Identifying tampered regions using singular value decomposition in digital image forensics. In: Computer Science and Software Engineering, 2008 International Conference on, IEEE, pp 926–930
15. Li G, Wu Q, Tu D, Sun S (2007) A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD. In: Multimedia and Expo, 2007 I.E. International Conference on, IEEE, pp 1750–1753
16. Li J, Li X, Yang B, Sun X (2015) Segmentation-based image copy-move forgery detection scheme. IEEE Trans Inf Forensics Secur 10(3):507–518
17. Lin H, Wang C, Kao Y (2009) Fast copy-move forgery detection. WSEAS. Trans Signal Process 5(5):188–197
18. Lowe DG (1999) Object recognition from local scale-invariant features. In: Computer vision, 1999. The proceedings of the seventh IEEE international conference on, IEEE, pp 1150–1157
19. Luo W, Huang J, Qiu G (2006) Robust detection of region-duplication forgery in digital image. In: Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, IEEE, pp 746–749
20. Mahdian B, Saic S (2007) Detection of copy–move forgery using a method based on blur moment invariants. Forensic Sci Int 171(2):180–189
21. Pan XY, Lyu S (2010) region duplication detection using image feature matching. IEEE Trans Inf Forensics Secur 5(4):857–867. doi:10.1109/Tifs.2010.2078506
22. Popescu AC, Farid H (2004) Exposing digital forgeries by detecting duplicated image regions. Dept Comput Sci, Dartmouth College, Tech Rep TR2004–515
23. Pun CM, Yuan XC, Bi XL (2015) Image forgery detection using adaptive over-segmentation and feature points matching. IEEE Trans Inf Forensics Secur 10(8):1705–1716
24. Ryu SJ, Kirchner M, Lee MJ, Lee HK (2013) Rotation invariant localization of duplicated image regions based on zernike moments. IEEE Trans Inf Forensics Secur 8(8):1355–1370. doi:10.1109/Tifs.2013.2272377
25. Sekhar R, Shaji RS (2016) A study on segmentation-based copy-move forgery detection using DAISY descriptor. In Proceedings of the International Conference on Soft Computing Systems. Springer, India, pp 223–233
26. Shivakumar B, Baboo LDSS (2011) Detection of region duplication forgery in digital images using SURF. IJCSI International Journal of Computer Science Issues 8(4):199–205
27. Wang J, Liu G, Li H, Dai Y, Wang Z (2009a) Detection of image region duplication forgery using model with circle block. In: Multimedia Information Networking and Security, 2009. MINES'09. International Conference on, IEEE, pp 25–29
28. Wang J, Liu G, Zhang Z, Dai Y, Wang Z (2009b) Fast and robust forensics for image region-duplication forgery. Acta Automat Sin 35(12):1488–1495

29. Xu B, Wang J, Liu G, Dai Y (2010) Image copy-move forgery detection based on SURF. In: Multimedia Information Networking and Security (MINES), 2010 International Conference on, IEEE, pp 889–892
30. Yu L, Han Q, Niu X (2014) Feature point-based copy-move forgery detection: covering the non-textured areas. Multimed Tools Appl 75(2):1159–1176. doi:10.1007/s11042-014-2362-y
31. Zhu Y, Shen X, Chen H (2015) Copy-move forgery detection based on scaled ORB. Multimed Tools Appl 75(6):3221–3233. doi:10.1007/s11042-014-2431-2

Xiuli Bi received her B.Sc. degree in application of electronic technology in 2004 and the M.Sc. degree in signal and information progressing in 2007, both from Shannxi Normal University, China. From 2007 to 2013, she was a lecturer in School of Computer Science, Beijing Institute of Technology, China. From 2013 to 2014, she was a research assistant in the University of Macau. She is currently working towards the Ph.D. degree in software engineering at the University of Macau. Her research interests include image processing; multimedia security and image forensics.

Prof. Pun received his B.Sc. and M.Sc. degrees in Software Engineering from the University of Macau in 1995 and 1998 respectively, and Ph.D. degree in Computer Science and Engineering from the Chinese University of Hong Kong in 2002. He is currently an Associate Professor and Head of the Department of Computer and Information Science of the University of Macau. He has investigated several funded research projects and published more than 100 refereed scientific papers in international journals, books and conference proceedings. Dr. Pun has served as the General Chair for the 10th & 11th International Conference Computer Graphics, Imaging and Visualization (CGIV2013, CGIV2014), and program/session chair for several other international conferences. He has also served as the editorial member/referee for many international journals such as IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Image Processing, Pattern Recognition, etc. His research interests include Digital Image Processing; Digital Watermarking; Pattern Recognition and Computer Vision; Intelligent Systems and Applications. He is also a Senior Member of the IEEE and a professional member of the ACM.

Dr. Yuan received her B.Sc. degree in Electronic Information Technology from the Macau University of Science and Technology in 2008, M.Sc. Degree in E-Commerce Technology and Ph.D. degree in Software Engineering from the University of Macau in 2010 and 2013 respectively. From 2014 to 2015, she was a postdoctoral fellow at the Department of Computer and Information Science of the University of Macau. She is currently an Assistant Professor at the Faculty of Information Technology of the Macau University of Science and Technology. Her research interests include Digital Image Processing; Digital Watermarking; Digital Audio/Video Processing; and Multimedia Forensics. She is also a Member of the IEEE.