




# Copy-move forgery detection using adaptive keypoint filtering and iterative region merging

Jun-Liu Zhong<sup>1</sup> · Chi-Man Pun<sup>1</sup> 

Received: 20 September 2018 / Revised: 1 April 2019 / Accepted: 21 May 2019 /  
Published online: 7 June 2019

© Springer Science+Business Media, LLC, part of Springer Nature 2019

## Abstract

Copy-move forgery detection can generally be divided into two categories: block-based or keypoint-based methods. However, the existing block-based methods are usually lack of efficiency and the keypoint-based methods have not good detection performance. In this paper, a novel method using the adaptive keypoint filtering and iterative region merging is proposed for copy-move forgery detection. First, a feature extraction algorithm is presented to obtain the candidate keypoint pairs. Subsequently, adaptive keypoint filtering involving adaptive nearest neighbor pair filtering and outlier filtering is proposed to remove the outliers and obtain the inlier (authentic keypoint) pairs. The iterative region merging involving adaptive region iteration and region merging is proposed to iteratively generate more neighboring keypoint pairs and then merge the image segmentations (superpixels) to implement the copy-move region matting. Compared with other state-of-the-art methods, a series of experiments show that the proposed method can overcome defects and achieve better efficiency while keeping the high detection precision in copy-move forgery detection even under conditions that include various post-processing distortions.

**Keywords** Adaptive keypoint filtering · Iterative region merging · Copy-move forgery detection

## 1 Introduction

In recent years, ordinary people have acquired easy access to powerful and easy-to-use image editing tools, such as Photoshop and so on. These convenient tools can add spice to people's lives. However, some malicious users abuse these powerful tools for their ends and tamper with images without the authors' permission. These forged images increasingly and negatively burdened society [26]. One of the most active subtopics in forgery detection involves copy-move forgery [9, 26], in which a part of an image is copied from single or multiple regions and then pasted into other parts of the same image to obscure important content. In general, copy-

---

✉ Chi-Man Pun  
cmpun@umac.mo

<sup>1</sup> Department of Computer and Information Science, University of Macau, Macau, SAR, China

move forgery operations are used not just in copying and moving (i.e., translation), but also includes geometrical distortions and post-processing operations such as scaling, rotation and compression. Detecting these imperceptible regions is difficult after such post-processing operations. Because the copy and paste regions are sourced from regions of the same image, their inner features are similar and compatible. Many proposed copy-move forgery detection (CMFD) methods have focused on correlating the inner features of detected regions to detect copy-move forgeries.

Based on previous research, CMFD methods can generally be divided into two categories [9, 35]: block-based [5, 11, 13, 14, 16, 17, 19–22, 24, 25, 30–32] or keypoint-based methods [1, 2, 4, 6–8, 10, 15, 18, 23, 27–29, 33]. The existing block-based and keypoint-based methods both employ similar processing procedures [9].

The main difference between block-based and keypoint-based methods lies in feature extraction. Block-based methods [9] divide the image into overlapped blocks that are typically rectangular, but some improved methods have proposed circular blocks [14, 35]. Then, various feature extraction algorithms have been applied to compute the feature vectors for each block. Subsequently, relevant detected regions are matched based on feature vector coefficients. Over the last fifteen years, numerous block-based methods have been proposed for CMFD. Fridrich et al. [13] first proposed the quantized Discrete Cosine Transform (DCT) coefficients to extract features of the overlapped and rectangular blocks. Popescu et al. [22] applied principal components analysis (PCA) to reduce the dimension of DCT. The  $8 \times 8$  DCT coefficient blocks [19], DCT transform Domain [20], the sum of the pixel intensities [21] and the histogram of orientated gradients [16] are proposed to extract block features. However, the extracted features of these methods are lack of spatial invariances. They do not work well in copy-move forgery detection. In recent years, to analyze intrinsic image features, some invariant moment methods have been proposed for CMFD. Ryu et al. [24] employed the magnitude of Zernike moments against rotation operation and constructed an algebraic rotation moment invariant. Ryu et al. [25] further proposed constructing copy-rotate-move (CRM) detectors for the overlapping blocks. Ustubioglu et al. [30] proposed calculating RGB color moments and entropy from the overlapping blocks. Yap et al. [31] proposed Polar Harmonic Transforms (PHTs). PHTs encompass orthogonality and invariance. Moreover, the kernels of PHTs are much simpler than are Zernike moments. Bi et al. [5] proposed the multi-level mask, then used Polar Complex Exponential Transform (PCET) that belongs to PHTs to extract features of the multi-level masks. The method has a good detection precision but in low efficiency. Gan et al. [14] and Li et al. [17] proposed the Discrete PCET (DPCET) and rotationally invariant Polar Cosine Transform (PCT) to extract block features, respectively. Emam et al. [11] also employed DPCET to extract features of the segmented blocks. Then, they used Locality Sensitive Hashing (LSH) to identify similar blocks. DPCET works well in detecting translation and rotation distortions and can precisely indicate the contour of copied regions. The LSH search also performs well in terms of accuracy but is not as fast as is desirable. There are two serious problems in block-based methods. The first one is incompetent to resist scaling distortions. The second one is due to its inefficiency. Zandi et al. [32] proposed an iterative procedure to adjust the density of keypoints; however, as with other block-based methods, this method also lacks scaling invariance.

To raise efficiency and handle the scaling distortion, the keypoint-based methods are proposed to extract the image features from the entire image. The popular keypoint-based methods are Scale Invariant Feature Transform (SIFT) [1, 4, 6, 8, 15, 18, 23, 29, 33] and Speeded Up Robust Features (SURF) [7, 10, 27]. SIFT and SURF methods are the feature

extraction algorithm of the image. However, both of them only show the extracted the local maxima and minima as keypoints located on the detected suspicious regions but fail in describing the contour of the regions. Therefore, they also fail to output satisfactory detection results. To compare the method performances, we should employ some post-processed procedures, such as filtering, classifying and matting, etc. to indicate the detected regions. Pun et al. [23] proposed Simple Linear Iterative Clustering (SLIC) to segment the image into superpixels. It provided an important clue for better image matting. But the fixed threshold of SLIC did not adaptively segment the superpixels accurate.

In this paper, we present a new copy-move forgery detection scheme using adaptive keypoint filtering and iterative region merging. The main contribution of the proposed method is listed as follows:

- 1) The adaptive keypoint filtering procedure is the first time to measure the classification errors of the extracted keypoints. It can correct the misclassified keypoints and then sharply reduce the classification error of keypoints. It can obtain as many inliers as possible to get the accurate affine matrices for the geometrical transformation evaluation of the forgery regions.
- 2) Iterative region merging is proposed to iteratively generate more keypoint (inlier) pairs and their suspected regions which are based on the invariant features and accurate affine matrices, then merge the adaptive superpixels to implement the copy-move region matting. The iteration and the merging algorithm can more precisely indicate the forgery regions, no matter single or multiple forgeries.

The remainder of this paper is organized as follows. The related work is described in Section 2. The proposed method using adaptive keypoint filtering and iterative region merging is presented in Section 3. Section 4 presents the experiments and their discussions for CMFD, and Section 5 provides concluding remarks and directions for future work.

## 2 Related work

In this section, we introduce some notable methods which are relevant to the proposed method. Pun et al. [23] have presented SIFT to extract the image features (keypoints) effectively. Each SIFT keypoint has 128-dimensional SIFT descriptors which contain localization information, gradient amplitude, dominant orientation, and scale. Due to the low extraction efficiency of SIFT, the improved local descriptor SURF [7, 10, 27] with only 64-dimensional descriptors uses the Fast Hessian to extract keypoints. Christlein et al. [9] have demonstrated that SURF descriptors have better feature extraction and detection performance than SIFT descriptors in copy-move forgery detection scheme. The efficiency of SURF is also better than SIFT. The local descriptors, no matter SIFT or SURF, are only some sparse keypoints. They cannot cover the whole forgery regions. Zandi et al. [32] proposed iterating the interest points to obtain the suspected regions, limiting the number of iterations to 4. The main defect of the method used in [32] is similar to that of the block-based method [11] which is unable to address scaling distortions-especially large-scale scaling. Pun et al. [23] proposed a similar iteration algorithm around the keypoints. It can resist the scaling operation. However, the iterative keypoints which are based on the RGB features do not have the rotation invariance [5], fail in resisting rotation attacks. Besides that, the iteration is a random operation that it is not clear how many

iterations of this is a satisfactory result. What is worse is that not all the keypoints are clustered into the correct classifications. It will lead to iteration errors. Li et al. [18] employed Simple Linear Iterative Clustering (SLIC) to segment the host image into meaningful blocks (superpixels). Then, when the number of inliers in every block satisfies the threshold, the corresponding superpixels pairs will be filled. However, this region-filling method has some defects as described below. First, the threshold of the initial segmentation with a fixed empirical value cannot get the satisfied segmentation. Second, SLIC is only an approximate texture segmentation method. It cannot find precise locations for some forged regions. Therefore, region-filling operations that rely only on SLIC segmentation with a fixed threshold do not obtain satisfactory matting results.

### 3 The proposed method

To overcome the defects of methods mentioned above, we propose a new copy-move forgery detection scheme which takes full advantage of the block-based and keypoint-based methods, namely efficiency of keypoint extraction and accuracy of the block/region filling. The proposed method contains three main stages: 1) keypoint extraction and matching (sub-section 3.1); 2) adaptive keypoint filtering (sub-section 3.2); 3) iterative region merging (sub-section 3.3).

The framework for the proposed method is depicted in Fig. 1. First, we employ the image pre-processing that contains the median filtering and color-to-gray conversion. Then, we implement three main processing stages as follow: 1) In the keypoint extraction and matching

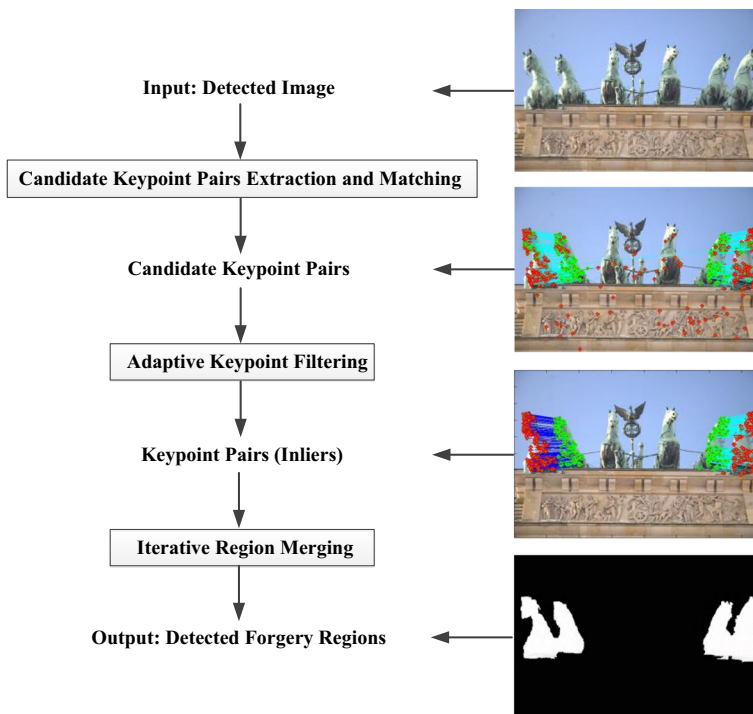


Fig. 1 The framework of the proposed CMFD scheme

stage, we present Speeded Up Robust Features (SURF) to extract the candidate keypoints and employ Best-Bin-First search and Nearest Neighbors test to match the candidate keypoint pairs. 2) Subsequently, the adaptive keypoint filtering stage involving adaptive nearest neighbor pair filtering and outlier filtering sub-stages is presented to remove the outliers and obtain the inlier (authentic keypoint) pairs. The first sub-stage adaptively removes the nearest-neighbor pairs by employing the Euclidean distance, while the second sub-stage evaluates the rest keypoints and then corrects misclassified keypoints to obtain as many inlier pairs as possible. Adaptive keypoint filtering can identify both single and multiple forgeries by adaptively employing inliers to cover them. 3) Finally, an iterative region-merging stage involving adaptive region iteration and region merging sub-stages is presented to obtain the forged regions. The first sub-stage proposes a random sample consensus (RANSAC) and the affine algorithm iteratively to generate the neighboring keypoints (NKs) for each corresponding inlier. Discrete Polar Complex Exponential Transform (DPCET) is employed to extract the circular block features corresponding to the above keypoints. The matching circular blocks generate the suspected regions. The segmentations (superpixels) which are segmented by Simple Linear Iterative Clustering (SLIC) are merged by iterating the suspected regions to determinate the copy-move regions.

### 3.1 Keypoint extraction and matching

This section describes and discusses the detailed steps for the feature (keypoint) extraction and matching algorithm. Detailed descriptions of the adaptive keypoint filtering and iterative region-merging algorithm, which are the two main contributions of this paper, are provided in Sub-sections 3.2 and 3.3, respectively.

Image feature extraction is an important task in CMFD. There are many types of feature extraction methods, such as SIFT [6], SURF [27], PCET [11] and so on. Based on the analysis of the above methods given in the introduction, we employ SURF methods [27] which is more effective than SIFT to extract the image features from the entire image. Then, the matching algorithm is employed to match the feature descriptors. The extracted SURF image features are expressed in the form of keypoints. Suppose there are a set of candidate keypoints  $P = \{p_1, p_2, \dots, p_n\}$  with their 64-dimensional SURF descriptors  $\{Sd_{i,1}, Sd_{i,2}, \dots, Sd_{i,128}\}$  in high-dimensional feature space; each candidate keypoint has 64-dimensional feature including rotation, scaling, orientation features, and so on. To distinguish different clusters and match candidate keypoints into pairs, various clustering methods are presented in [9]. The kd-tree algorithm [15, 18] and Best-Bin-First (BBF) [3] are the commonly used methods for obtaining the approximate nearest neighbors. In our proposed method, we employ Euclidean distance to measure the correlation between two candidate keypoints. To intuitively compare the correlation between two candidate keypoints, their distance ratio is used as the estimated standard. A 2NN test [1] is normally employed as a ratio threshold when matching two candidate keypoints as a pair. The list of the Euclidean distances between the  $i^{th}$  keypoint and the other keypoints is sorted in ascending order to identify similar feature vectors. We use a powerful tool, like OpenCv, which can provide the FlannBasedMatcher function, for implementing the SURF feature matching easily. The equation is given in (1), and the 2NN test is given in (2).

$$CD_i = (cd_{i,1}, cd_{i,2}, \dots, cd_{i,n}) \quad (1)$$

where  $cd_{i,1}$  is the correlation distance of the closest neighbor between the  $no.i$  candidate

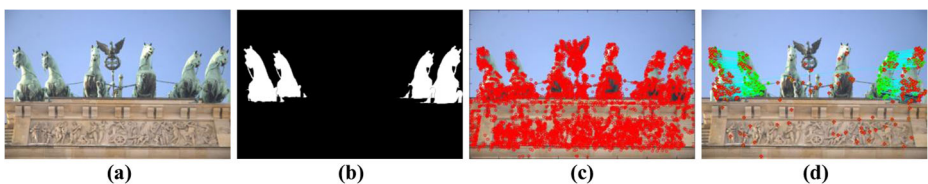
keypoint and other keypoints, namely, the minimum Euclidean distance, while  $cl_{i,2}$  is the second-closest neighbor. To perform a more effective matching method, Amerini et al. [1] suggested using the ratio of  $cd_{i,1}$  to  $cd_{i,2}$  to match the candidate keypoint pairs. Each matched pair is normalized with Eq. (2):

$$cl_{i,1}/cl_{i,2} < T_1, \text{ where } T_1 \in (0, 1) \quad (2)$$

The number and distribution of the keypoints are also important. When  $T_1$  is close to 1, the 2NN test can obtain more candidate keypoint pairs. Otherwise, the small number of the extracted keypoint pairs fails to detect the forged regions. A good matching algorithm should find as many keypoints as possible to concentrate and cover the copy-move regions. Others, namely, mismatched point pairs, should be kept to a minimum. Therefore, we suggest fixing  $T_1$  to 0.5 based on empirical findings. Figure 2 shows the process of the candidate keypoint extraction and matching algorithm. After keypoint matching is complete, the goal was that the matched keypoint pairs should be concentrated in the copy-move regions. It can be observed from Fig. 2-(d) that there are some false keypoint pairs. Therefore, the adaptive keypoint filtering to obtain the inliers is given a more detailed description in Sub-section 3.2.

### 3.2 Adaptive Keypoint filtering

Various filtering algorithms have been proposed to remove weak and false keypoint pairs (matches). Euclidean distance and correlation coefficients are the most common filtering methods [9]. However, the fixed threshold involved in the Euclidean distance and correlation coefficient filtering methods limits their ability to obtain ideal filtering results. In this section, the adaptive filtering algorithm, which fuses the state-of-the-art algorithms, is proposed to enable better keypoint filtering. Figure 3 shows a flowchart of the adaptive keypoint filtering method that is divided into two sub-stages: adaptive nearest neighbor pair filtering and adaptive outlier filtering. The purpose of the first sub-stage is to remove the nearest neighbor pairs and obtain as many the correct keypoint pairs as possible. First, Euclidean distance with a fixed threshold is employed to filter out the neighboring keypoints pairs in which both members are close to each other. Then, a new filtering threshold based on average distance of the remaining keypoint pairs and the low-frequency distribution  $P_{LF}$  of an image is employed to adaptively remove the nearest neighbor keypoint pairs once again. After the first sub-stage filtering, most of the unwanted outliers have been removed. Then, the second filtering sub-stage is undertaken to adaptively correct misclassified keypoints and address the multiple forged regions. Random sample consensus (RANSAC) is proposed to repeatedly evaluate the keypoint clusters and obtain the inliers, remove outliers. Finally, the authentic inliers of the  $i^{\text{th}}$  cluster are obtained. After the second filtering sub-stage, the inliers are preserved and the outliers are removed.



**Fig. 2** The process of candidate keypoint extraction and matching: (a) shows the host image, (b) is the ground-truth region, (c) illustrates the extracted maxima and minima, and (d) illustrates the candidate keypoint pairs of the image

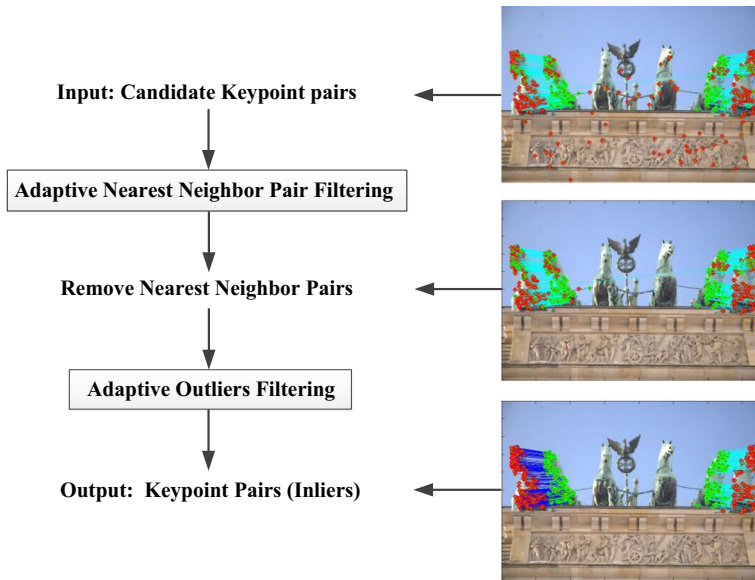


Fig. 3 A flowchart of the adaptive keypoint filtering process

### 3.2.1 Adaptive nearest neighbor pair filtering

Euclidean distance [9] is a commonly used method for filtering out mismatched keypoint pairs, especially for the nearest neighbor pairs. The Euclidean distance of the  $no.i$  keypoint pairs is defined as follows.

$$Ed_i = \|m_{i,1}, m_{i,2}\|_2 > T_b, \tag{3}$$

where  $\| \cdot \|_2$  is the L-2 norm,  $Ed_i$  represents the  $i^{th}$  candidate keypoint pairs, and  $m_{i,1}$  and  $m_{i,2}$  are the two candidate keypoints in a single matched pair. Here,  $T_b = (H + W)/200$  is the filtering distance, where  $H$  and  $W$  are the height and width of the detected image, respectively. The 1st Euclidean distance is employed to filter out the nearest neighbor matches. Then, the adaptive threshold of the Euclidean distance is applied to the 2nd filtering. The weight of the adaptive filtering distance is a sensitive research point. Some initial threshold of image segmentation methods can provide important cues. Pun et al. [23] proposed an adaptive over-segmentation algorithm to segment the image into non-overlapped blocks. The segmentation algorithm is based on the four-levels of Discrete Wavelet Transform (DWT), using the ‘Haar’ wavelet. Zheng et al. [34] employed the Haar wavelet to set the initial size of the segmentation. The candidate keypoint pairs in the same block are considered as mismatched pairs and the initial size is considered as the threshold for the Euclidean distance [34]. The Haar coefficient is defined as follows:

$$E_{LF} = \sum |CA_4| \tag{4}$$

$$E_{HF} = \sum_i (\sum |CD_i| + \sum |CH_i| + \sum |CV_i|) \tag{5}$$

$$P_{LF} = \frac{E_{LF}}{E_{LF} + E_{HF}} \times 100\%, \tag{6}$$

where  $i = 1, 2, 3, 4$  and  $CA_4$  denotes the approximated coefficients at the 4th level of DWT, while  $CD_i, CH_i,$  and  $CV_i$  denote the detailed coefficients at the  $i^{th}$  level of DWT.  $P_{LF}$  is a reflection of the image low-frequency distribution. When  $P_{LF}$  is close to 1, the energy of the detected image is strongly concentrated in the low-frequency band or the low resolution. When the speed of change is low and smooth, the distribution of the keypoint pairs may be more widely spread and the number of the keypoint pairs is more likely to be sparse or fewer. Therefore,  $P_{LF}$  is near to 1, and the filtering threshold is set to a higher distance. Pun et al. [23] and Zheng et al. [34] recommended using the above setting. These methods provide a good scheme for adaptive segmentation but consider only the frequency distribution to segment the image. The threshold is also set to a fixed value based on empirical findings. In Fig. 4-(a3), the candidate keypoint pairs are all filtered out by employing the threshold setting used in [23]. There are no keypoint pairs in the detected image; therefore, the detection algorithm failed. A fixed threshold that either too large or too small does not achieve satisfactory results. Therefore, we propose an adaptive threshold to obtain improved filtering results. The average distance is considered as another factor for adaptive segmentation. The average Euclidean distance of the remaining keypoint pairs is defined as follows.

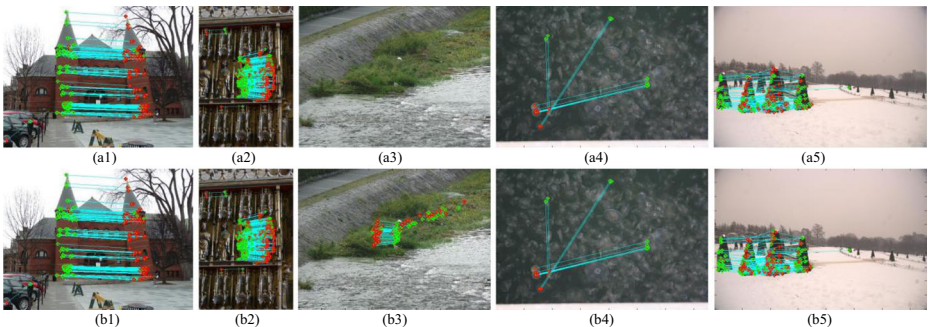
$$Edp = \frac{1}{m} \sum_{i=1}^m \|p_{i,1}, p_{i,2}\|_2, \tag{7}$$

where  $Edp$  is the average distance of all matched pairs, and  $p_{i,1}$  and  $p_{i,2}$  are the candidate keypoints of the  $no.i$  pair.

The adaptive algorithm contains two steps. Figure 4 shows an example of the filtering algorithm. First, the nearest neighbor pairs are filtered using the low threshold. Then, the average distances of the remaining keypoint pairs are calculated. The average distance  $Edp$  with  $P_{LF}$  as the weight-coefficient is used to adaptively filter the candidate keypoint pairs. The adaptive filtering threshold is defined as (8):

$$T_p = P_{LF}^{4/3} \times Edp, \tag{8}$$

The images in Fig. 4-(a1)~(a5) show the filtering results of the fixed filtering algorithm [23, 34]. Figure 4-(b1)~(b5) show the filtering results of the adaptive nearest neighbor filtering algorithm. Note that Fig. 4-(a1), (a2), (a4) and (a5) have similar filtering performances compared to Fig. 4-(b1), (b2), (b4) and (b5), respectively. In Fig. 4-(a3), the fixed threshold



**Fig. 4** The comparison results for the filtering algorithm. The top row shows the filtering results of the fixed filtering algorithm [23, 34], while the bottom row shows the filtering results of the adaptive nearest neighbor filtering algorithm



will filter out all the keypoint pairs. The size of the image in Fig. 4-(a3) is  $3888 \times 2592$  and  $P_{LF} = 0.4075$ . In Fig. 4-(b3), it can be observed that the adaptive filtering algorithm removes the nearest neighbor pairs effectively while preserving other correctly matched pairs. Compared to other fixed algorithms, filtering the candidate keypoints using the proposed filtering algorithm can cover the copy-move regions more precisely.

### 3.2.2 Adaptive outliers filtering

The first filtering stage is simply a preliminary filtering step in the scheme. After filtering, some unwanted outliers remain, as shown in the top left of Fig. 4-(b2). To eliminate the effect of these unwanted outliers, random sample consensus (RANSAC) [9, 12], the state-of-the-art method, is introduced to find an affine matrix  $H$  to estimate the best correlation coefficient among a certain number of trials. The goal is to filter out the falsely matched pairs (outliers) to obtain the correctly matched pairs (inliers). RANSAC has been shown to perform well in filtering operations for single forgeries, such as in Fig. 4-(b1) and (b2). However, RANSAC is not suitable for correctly filtering multiple copy-move forged regions. As shown in Fig. 4-(b1)–(b5) and Fig. 5-(a), another serious problem is that the keypoint classifications obtained by employing SURF and 2NN test are not entirely correct. Some candidate keypoints rightly belong to the  $A$  class but are assigned to the  $B$  class of the same cluster. The candidate keypoints belonging to the  $A$  class are denoted in green, while the candidate keypoints of the  $B$  class are denoted in red in Fig. 5-(a). Therefore, the classification and filtering must be re-estimated using RANSAC and the affine matrix to correct the misclassifications of the keypoint pairs. First, we employ the 1st RANSAC to filter out the candidate outliers and obtain the estimated candidate inliers used to obtain the affine matrix  $H_1$ . Then, the candidate outliers are re-evaluated by employing  $H_1$  to measure whether these outliers truly belong to the inliers. When these outliers have been misclassified, they must be corrected to become inliers. This step eliminates misclassifications and obtains the  $A_1$  class and  $B_1$  class of the 1st correct cluster (inliers). The  $A_1$  class and  $B_1$  class of the 1st correct cluster are shown in Fig. 5-(b). When multiple copy-move forged regions exist, the rest of the outliers are re-estimated as the sources of the 2nd keypoint cluster. When the number of non-collinear outlier pairs falls below 3, the re-estimation algorithm is terminated. Otherwise, the cluster filtering is continually updated using the iteration steps described above. When the image contains only one forged region, the number of iterations is 1. The adaptive filtering method fuses the affine transform and RANSAC to obtain superior performance. Figure 5 shows the steps of the keypoint filtering algorithm. It can be observed that a portion of the candidate keypoint pairs in Fig. 5-(a) are misclassified. Figure 5-(b) shows the  $A_1$  class and the  $B_1$  class from the adjusted 1st clusters, namely, the 1st correct inliers. Figure 5-(c) shows the adjusted 2nd clusters, namely, the 2nd correct inliers. Figure 5-(d) shows the two correct clusters in the same image.



**Fig. 5** The image output of the steps of the adaptive outliers filtering algorithm

To conduct the second stage filtering, the RANSAC is employed to estimate the results of clusters. The number  $k$  for each RANSAC is defined as follows:

$$k = \log(1-p)/\log(1-w^m). \tag{9}$$

where the number  $k \leq 200$ , the confidence  $p$  is set to 0.995,  $w$  is the inliers ratio of all estimated pairs,  $m$  is the number of the estimated samples, and  $m > 3$ . There are two problems when using RANSAC. The first is that multiple copy-move regions may exist. RANSAC can obtain the 1st inliers but abandons the rest as outliers, meaning that the 2nd inliers or other inliers will be not obtained. The second problem is that some candidate keypoints are incorrectly classified by SURF extraction. These incorrect classifications will be regarded as outliers and abandoned. Smaller-sized regions with a limited number of candidate keypoint pairs cannot be easily detected. To obtain more inliers for the estimation and matting, the results of RANSAC should be analyzed. The affine matrix  $H_1$  can be easily obtained by calculating the inliers.

$$\begin{bmatrix} X'_1 \\ Y'_1 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ 1 \end{bmatrix} = H_1 \begin{bmatrix} X_1 \\ Y_1 \\ 1 \end{bmatrix}, \tag{10}$$

where  $X_1 = [x_{1,1}, x_{1,2}, \dots, x_{1,m_1}]$  and  $Y_1 = [y_{1,1}, y_{1,2}, \dots, y_{1,m_1}]$  are the coordinates of the class  $A_1$  of the inliers obtained by the 1st RANSAC,  $m_1$  is the number of candidate inlier pairs,  $X'_1 = [x'_{1,1}, x'_{1,2}, \dots, x'_{1,m_1}]$  and  $Y'_1 = [y'_{1,1}, y'_{1,2}, \dots, y'_{1,m_1}]$  are the coordinates of class  $B_1$ . According to the least-squares method, the affine matrix  $H_1$  can be obtained as follow.

$$H_1 = \left[ \left[ [X_1 \ Y_1 \ 1]^T [X_1 \ Y_1 \ 1] \right]^{-1} [X_1 \ Y_1 \ 1]^T [X'_1 \ Y'_1 \ 1] \right]^T. \tag{11}$$

Then, the 1st candidate outliers are evaluated as to whether they belong to the misclassified keypoints. The coordinates of the keypoints of the outliers in a pair are exchanged to adjust possible misclassifications. The inverse transform of the affine matrix  $H_1$  can be employed to estimate the relationship between the inliers and the adjusted outliers. The exchanged coordinates of outliers are measured as follows.

$$\mu = \left| \begin{bmatrix} X_2 \\ Y_2 \\ 1 \end{bmatrix} - H_1^{-1} \begin{bmatrix} X'_2 \\ Y'_2 \\ 1 \end{bmatrix} \right| = \left| \begin{bmatrix} X_2 \\ Y_2 \\ 1 \end{bmatrix} - H'_1 \begin{bmatrix} X'_2 \\ Y'_2 \\ 1 \end{bmatrix} \right| = \left| \begin{bmatrix} X_2 \\ Y_2 \\ 1 \end{bmatrix} - \begin{bmatrix} h'_{11} & h'_{12} & h'_{13} \\ h'_{21} & h'_{22} & h'_{23} \\ h'_{31} & h'_{32} & h'_{33} \end{bmatrix} \begin{bmatrix} X'_2 \\ Y'_2 \\ 1 \end{bmatrix} \right| = \begin{bmatrix} \mu_{1,1} & \dots & \mu_{1,m_2} \\ \mu_{2,1} & \dots & \mu_{2,m_2} \\ \mu_{3,1} & \dots & \mu_{3,m_2} \end{bmatrix}, \tag{12}$$

where  $X_2 = [x_{2,1}, \dots, x_{2,m_2}]$  and  $Y_2 = [y_{2,1}, \dots, y_{2,m_2}]$  are coordinates of the class  $A'_1$  of the 1st candidate outliers, The coordinates of the outliers of the  $A'_1$  class and the  $B_1$  class are exchanged, where  $X'_2 = [x'_{2,1}, \dots, x'_{2,m_2}]$ ,  $Y'_2 = [y'_{2,1}, \dots, y'_{2,m_2}]$  are coordinates of the class  $B'_1$ ,  $m_2$  is the number of candidate outlier pairs, and  $\mu_{1,j}$  and  $\mu_{2,j}$  respectively represent the errors of the x-coordinate and y-coordinate. Here,  $\mu_{i,j}$  represents the distance between the exchanged coordinates of the  $no.j$  outlier to its unchanged coordinate,  $H_1^{-1} = H'_1$ ,  $\eta_{2,j} = (\mu_{1,j} + \mu_{2,j}) < \varepsilon = 8$ . This can be expressed in another form:

$$\eta_{2,j} = \left| x_{2,j} - \frac{h'_{11}x'_{2,j} + h'_{12}y'_{2,j} + h'_{13}}{h'_{31}x'_{2,j} + h'_{32}y'_{2,j} + h'_{33}} \right| + \left| y_{2,i} - \frac{h'_{21}x'_{2,j} + h'_{22}y'_{2,j} + h'_{23}}{h'_{31}x'_{2,j} + h'_{32}y'_{2,j} + h'_{33}} \right| \leq \varepsilon \tag{13}$$

where  $j$  represents the  $no.j$  exchanged outlier, if it satisfies the threshold. Specifically, this

outlier pair satisfies the  $H_1^{-1}$  affine transform, when the two candidate keypoints of an outlier pair keep the exchanged coordinates, namely, point  $A'_1(x_{2,j}, y_{2,j})$  is exchanged to point  $B'_1(x'_{2,j}, y'_{2,j})$  and classified to class  $B_1$ , and point  $B'_1(x'_{2,j}, y'_{2,j})$  is exchanged to point  $A'_1(x_{2,j}, y_{2,j})$  and reclassified as class  $A_1$ . Others pairs that do not satisfy the threshold will keep their original coordinates. Figure 5 shows the exchange process. In Fig. 5-(b), the  $A_1$  and  $B_1$  classes are denoted as green and red color points, respectively. Others are regarded as the 1st outliers. All the outliers are re-estimated and redistributed to the correct classes. Then, the 1st redistributed inliers are estimated to modify an affine matrix  $\vec{H}_1$ .

When a copy-move image exists in multiple forged regions, the iterative loop will continue until it meets the termination condition defined earlier. The filtered processing of multiple forgeries is shown in Fig. 5. The adaptively filtered algorithm accurately distinguishes the inliers.

### 3.3 Adaptive Region Iteration & Merging

Fig. 5-(d) shows the copy-move regions covered by the filtering keypoints (inliers). However, most of the inliers only roughly describe the suspected regions without exact region matting. Therefore, the region-filling algorithm must be performed to indicate the copy-move regions more clearly. Therefore, we proposed an iterative region-merging algorithm that contains adaptive region iteration and region merging for region filling. Figure 6 shows a flowchart of the iterative region-merging algorithm.

To precisely describe the contours and contents of the forged regions, a high-density of matched inlier pairs are needed to cover the forged regions. First, the inlier pairs are loaded as labeled keypoints (LKs). The 8 neighboring keypoints (NKs) of the LKs belonging to the  $A_1$

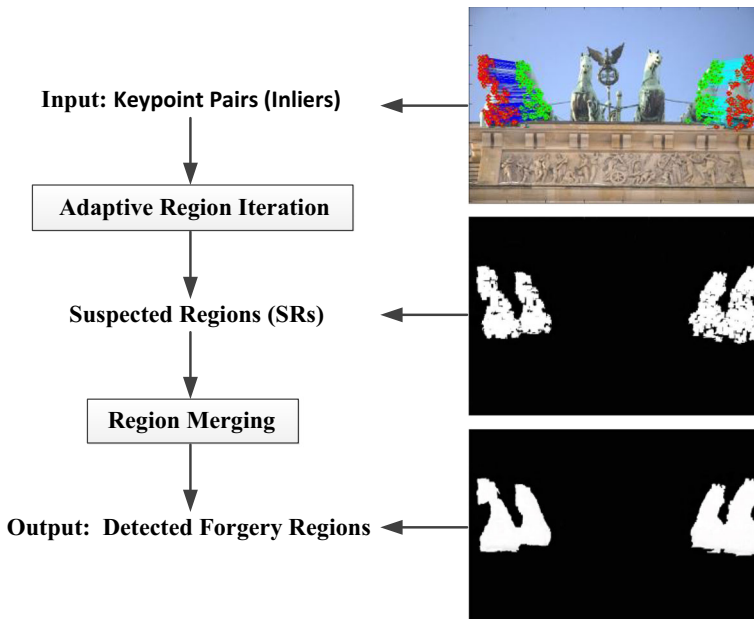


Fig. 6 The framework of the iterative region-merging algorithm

class are generated. The affine matrix obtained from sub-section 3.3.2 is employed to calculate the 8 NKs of each matched LK of the  $B_1$  class. The LKs of the  $A_1$  and  $B_1$  classes are respectively denoted as green and red color points in Fig. 5-(b). The labeled keypoints of  $A_1$  and  $B_1$  classes represents 1st clusters ( $LK_1$ ). The circular block of each LK and NK is calculated with DPCET to determine whether the NK pairs match. The circular block of the matched NK is filled to generate the 1st suspected regions. Then the 1st matched NKs are loaded as new 2nd labeled keypoints. The preceding steps are repeated iteratively until the termination condition is reached. Second, to accurately display the copy-move forged regions, a morphological operation, SLIC with an adaptively initial threshold, is employed to segment the host image into superpixels. Then, the pixel percentage of suspected regions to the corresponding superpixels is calculated to measure whether the ratio satisfies the criterion. Finally, the superpixels are not only used to fill the whole but also merged with suspected regions to locate the copy-move regions accurately.

### 3.3.1 Adaptive region iteration

Assume that  $LK_i = \left\{ \left( LK_{i,1}, LK'_{i,1} \right), \dots, \left( LK_{i,m_i}, LK'_{i,m_i} \right) \right\}$ ,  $LK_i$  represents the labeled keypoint pairs of the  $i^{th}$  cluster,  $LK_{i,j}$  and  $LK'_{i,j}$  are the  $A_i$  class and  $B_i$  class of the  $i^{th}$  cluster, respectively. An illustration is shown in the upper part of Fig. 6, where  $i = 1, 2, \dots, n$ , and  $n$  is the number of clusters, and  $j = 1, 2, \dots, m_i$  is the number of keypoint pairs of the  $i^{th}$  cluster. The 1st neighboring keypoints ( $NK_{1,j}$ ) of labeled keypoints ( $LK_{1,j}$ ) are defined as shown in (14).

$$NK_{1,j} = \cup NK_{1,j,\theta}, \tag{14}$$

where  $\theta = (0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ)$ , and the distance between the  $LK_{1,j}$  and  $NK_{1,j,90}$  is  $r$ . Equation (10) is employed to calculate the NKs coordinates of the  $B_i$  class that correspond to the NKs of the  $A_i$  class. The radius  $r$  of the circular block is a multiple of 10. The definition of  $r$  is given as follows:

$$r = \begin{cases} \text{INT}(M \times N / 10^6) \times 10, & \text{others} \\ 10, & M \times N < 10^6 \end{cases}, \tag{15}$$

where  $M$  and  $N$  are the dimensions of the host image. The circular block is shown in Fig. 7. The center pixel of the circular block represents the corresponding labeled keypoint (LK) or neighboring keypoint (NK). When the calculated values of the NK pairs matched each other, the circular block is filled to generate a suspected region. Emam et al. [11] and Gan et al. [14] proposed using Discrete Polar Complex Exponential Transform (DPCET) to extract the rotation features of the image. DPCET is an algorithm effective against rotation distortions, but it fails to detect scaling operations. To detect scaling operations, DPCET is employed to calculate the features of the circular block with a variable radius [14, 35]. DPCET is defined as follows [11]:

$$M_{kl} = \frac{4}{\pi r^2} \sum_{y=-r}^r \sum_{x=-r}^r \left[ \exp(i \times (2\pi k r^2 + \theta)) \right]^T f(x, y), \quad \|x, y\|_2 \leq r, \tag{16}$$

$$|M_{kl}| = |M_{kl}^{ROT}|, \tag{17}$$

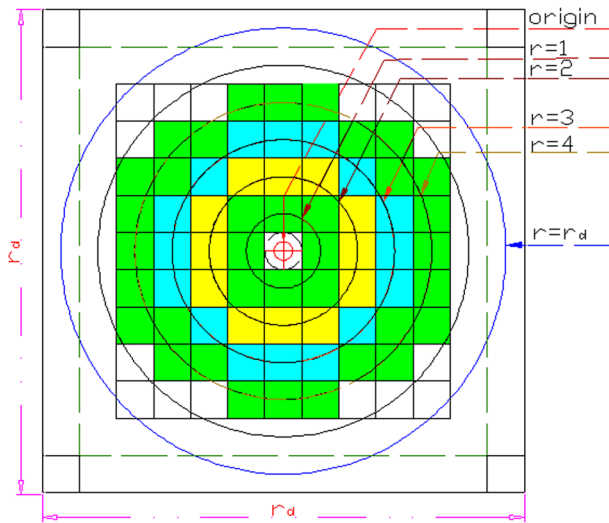


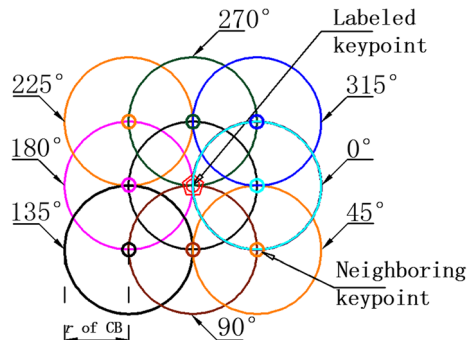
Fig. 7 The circular block example

where  $M_{kl}$  is the DPCET with  $k^{th}$  order and  $l^{th}$  repetition,  $\theta = \arctan(y/x)$ ,  $\|x, y\|_2 \leq r$ ,  $r$  is the radius of the circular block and  $M_{kl}^{ROT}$  represents the DPCET coefficients of the rotation operation. Equation (16) gives the rotation invariant for extracting the rotation features of the circular block. Equation (15) defines the circular block radius of the  $A_i$  class. To better calculate the circular block feature of the corresponding to  $B_i$  class, the scaling dimension is defined as  $\lambda$ , where  $\lambda$  is equal to mean value of  $h_{11}$  and  $h_{22}$  in (10). When  $\lambda$  is not greater than 0.7, the  $\lambda$  is scale-invariant to prevent the calculated errors of the too small circular block. When  $\lambda$  is greater than 0.7, the initial radius  $r$  of the circular block is defined as shown in (18).

$$r = \text{Int}(10 \times \lambda), \tag{18}$$

A circular block example and an illustration of neighboring keypoint (NK) are shown in Fig. 7 and Fig. 8, respectively. The Eq. (14) yields the 1st NK of each keypoint. Equations (15)–(18) provide the extracted geometrical features of the

Fig. 8 The illustration of a labeled keypoints and 8 neighboring keypoints



circular block. The local color feature of the corresponding circular block is calculated using (19) and (20).

$$M\_NK_{1,j,\theta} = 0.3R(NK_{1,j,\theta}) + 0.59G(NK_{1,j,\theta}) + 0.11B(NK_{1,j,\theta}), \tag{19}$$

$$M\_NK'_{1,j,\theta} = 0.3R(NK'_{1,j,\theta}) + 0.59G(NK'_{1,j,\theta}) + 0.11B(NK'_{1,j,\theta}), \tag{20}$$

where  $R()$ ,  $G()$ , and  $B()$  respectively denote the red, green and blue components of the corresponding circular block,  $M\_NK_{1,j,\theta}$  and  $M\_NK'_{1,j,\theta}$  are the RGB feature of the neighboring keypoint (NK) in the  $A_1$  and  $B_1$  class, and 1 means the 1st NK. The circular block of each NK pair will be filled when they meet the criterion defined in (21):

$$|M\_NK_{1,j,\theta} - M\_NK'_{1,j,\theta}| < T_{NK} \times |M\_NK_{1,j,\theta}|, \tag{21}$$

where  $T_{NK}$  is the threshold to measure the similarity between the compared NK pair. This paper proposes that  $T_{NK}$  be set 0.04 based on the experiments.

Assume that 1st suspected region denotes the  $A_1$  and  $B_1$  classes of the 1st effective neighboring keypoints (NK<sub>1</sub>), that the features of the circular block satisfy (19) and (20), and that  $i = 1, 2, \dots, n_1$ , where  $n_1$  is the number of NK<sub>1</sub>. Then, repeat the above steps to achieve the optimal region matting. The NK is iterated until it satisfies the termination condition of Eq. 22. It is noted that the neighboring keypoints located in the filled blocks of the other keypoints, did not need to repeat the calculation.

$$Termination = \frac{no.i^{th} NK}{no.(i + 1)^{th} NK} < 1, \tag{22}$$

It can be observed from Fig. 6 that the suspected regions cover the ground-truth regions precisely.

### 3.3.2 Region merging

To visually display the copy-move forged regions and restrict the suspected regions of the neighboring keypoint (NK), a morphological operation, Simple Linear Iterative Clustering (SLIC), is employed to segment the image into superpixels. The  $T_p$  in (8) is an adaptive coefficient based on calculating the distribution of keypoints and is used as the initial segmentation coefficient of SLIC. Then, the pixel percentage of the suspected region to the corresponding superpixel is calculated to measure whether the ratio satisfies the criterion. Finally, the superpixels and suspected regions are merged to fill the regions in three modes: the superpixel is completely filled, completely abandoned, or the suspected region preserves the pixels in its superpixel. These three filling modes are employed to indicate the detected region more accurately. The filling criterion is shown in (23).

$$If\ Percentage \begin{cases} > 70\%, \text{ Fill the whole superpixel} \\ < 20\%, \text{ Abandon the superpixel} \\ \text{others, Keep neighboring keypoints in superpixel} \end{cases}, \tag{23}$$

When the pixel percentage of the suspected region to the corresponding superpixel is over 70%, the entire superpixel is filled. When the pixel percentage of the suspected region to the

corresponding superpixel is below 20%, the superpixel is abandoned. In the other case, the pixels of the suspected region will be preserved in its superpixel. Some small holes and isolated pixels are also eliminated by employing mathematical morphological operations. The superpixels are merged with the suspected regions to implement the copy-move region-filling operation as shown in Fig. 10-(a1)~(a5).

## 4 Experiments and analysis

In section 4, a wide variety of experiments are conducted to evaluate the performances of the proposed method and the state-of-the-art methods under the geometric transform and multiple region forgeries.

### 4.1 Evaluation criteria

In our experiments, to evaluate the performance of the compared CMFD methods, we use two main parameters, *precision* and *recall* [11, 18, 23, 27] as the two criteria to analyze of the experimental results. *Precision* and *recall* are defined in Eqs. (24) and (25), respectively.

$$Precision = T_p / (T_p + F_p), \quad (24)$$

$$Recall = T_p / (T_p + F_n), \quad (25)$$

Using (24) and (25), *precision* and *recall* are employed to test the CMFD methods at both image and pixel levels. The image-level evaluation distinguishes the performance of the method in detecting overall image forgeries, while the pixel-level evaluation is localized to detect the performance at the forged region area.  $T_p$  represents True Positive. At the image level,  $T_p$  represents a forged image that is correctly identified. At the pixel level,  $T_p$  represents that the correct number of detected copy-move pixels were detected as forged pixels.  $F_p$  means False Positive. At the image level,  $F_p$  represents detection errors in which a real image or authentic region was incorrectly detected as a forgery. In pixel level,  $F_p$  represents the ratio of authentic pixels erroneously detected as forged pixels.  $F_n$  means False Negative. At the image level,  $F_n$  represents undetected forged images or regions incorrectly detected. At the pixel level,  $F_n$  represents the proportion of forged pixels that are undetected. To comprehensively measure the performance of the CMFD methods, the  $F_1$  score combines both *precision* and *recall*:

$$F_1 = 2 \times \frac{Precision \times recall}{precision + recall} \quad (26)$$

The closer  $F_1$  is to 1, the better the performance obtained by the CMFD method is.

### 4.2 Benchmark database for CMFD evaluation

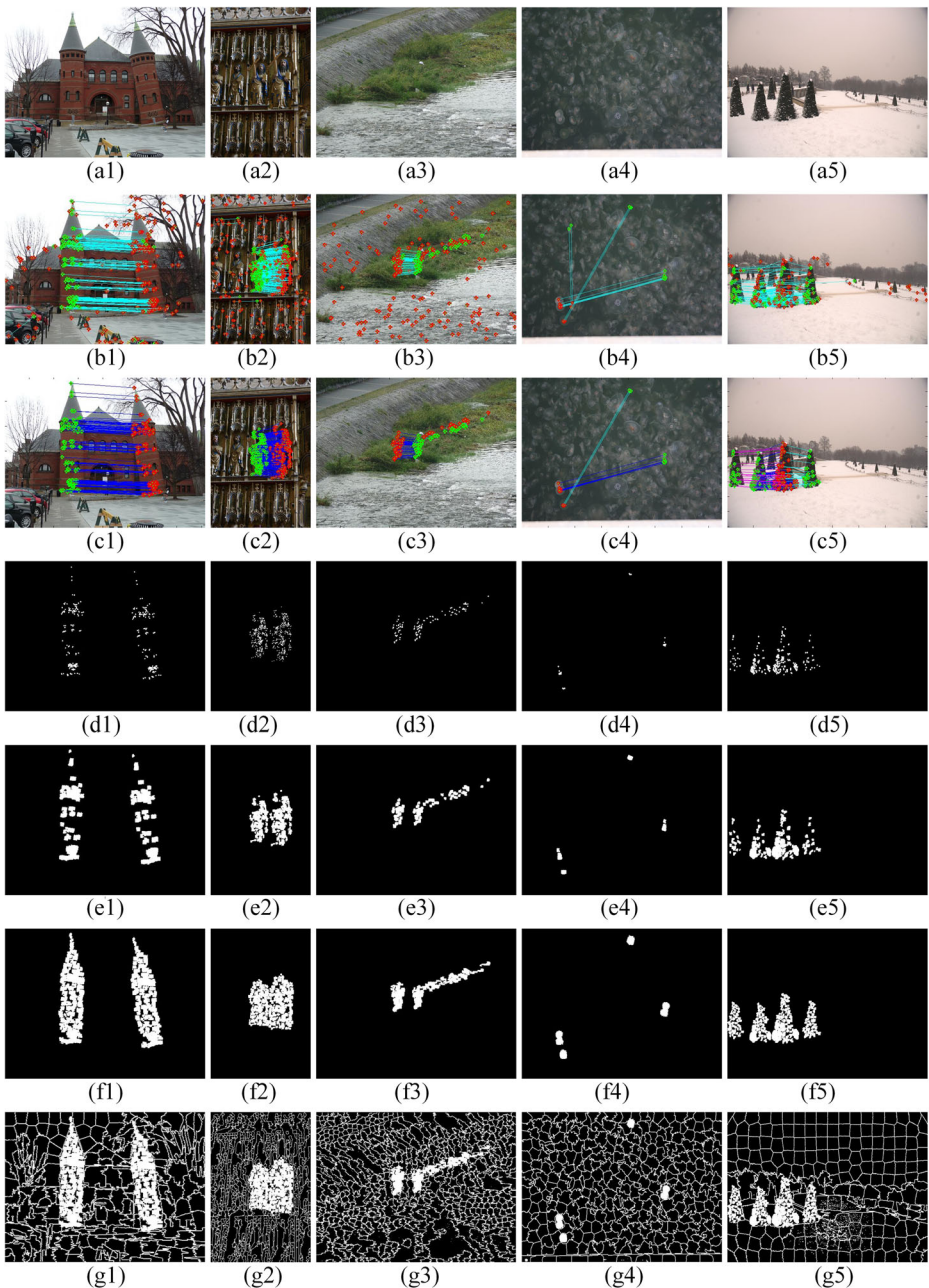
Standard benchmark databases are used as uniform assessment criteria to compare the performance of different CMFD methods. The benchmark databases used here were compiled by the Department of Computer Science at Friedrich-Alexander University [9]. The basic dataset is composed of 48 high-resolution base images as well as copied and pasted snippets

from these images to create copy-move forged images. The benchmark dataset contains rotated copies, scaled copies, down-sampled copies, splices with JPEG image compression, and so on. In our experiments, the existing state-of-the-art block-based method [11], keypoint-based method [1, 18, 23, 27] and the iterative interest-point method [32] were all tested to evaluate their performances. Figure 9 depicts the process used in the proposed method and Fig. 10 shows the detected results for the proposed method and the compared methods [11, 18]. The copy-move images contain several types of objects such as plants, animals, man-made objects and combinations of these. Figure 9-(a1)~(a5) shows the copy-move host images. Figure 9-(a1) shows the red tower image where the copied portion is rotated by  $10^\circ$ . Figure 9-(a2) shows the wood carvings image with a scaled-up 20% distortion. Figure 9-(a3) shows the fisherman image that contains multiple copy-move forged regions implemented by scaled-down 20% distortions. Figure 9-(a4) shows the jellyfish image with multiple copy-move regions in which each forged region is implemented a  $20^\circ$  rotation. Figure 9-(a5) shows the Christmas hedge image with multiple copy-move regions each of which is implemented by scaled-down 20% distortions. Figure 9-(a1) and (a2) show the single forged region. Figure 9-(a3) shows two separated copy regions corresponding to the two different forged regions. Figure 9-(a4) shows three separated copy regions corresponding to the three different forged regions respectively. Figure 9-(a5) shows one copy region corresponding to three forged regions. Figure 9-(b1)~(b5) shows the candidate keypoint pairs using a matching threshold of 0.5. Figure 9-(c1)~(c5) shows the results of the adaptive keypoint filtering. Figure 9-(d1)~(d5), (e1)~(e5) and (f1)~(f2) show the 1st, 3rd and ultimate iteration results of suspected regions, respectively. Figure 9-(g1)~(g5) show the relationship between the ultimate suspected regions and the superpixels. Figure 10-(a1)~(a5), (b1)~(b5), (c1)~(c5) shows the detected forgery results of the proposed method, the methods from [11, 18], respectively. Figure 10-(d1)~(d5) shows the ground-truth regions corresponding to the images in Fig. 9-(a1)~(a5), respectively. From the results shown in Fig. 10, it can be observed that our proposed method (shown in (a1) to (a5)) can achieve much better results. Figure 10-(a1)~(a3) and (a5) show that the matching between the iterative region-merging areas and the ground-truth areas can reach 90%. Figure 10-(a4) shows the correct region filling that occurred on the two forged regions, but the method missed the third forged region. It is shown in Figure 10-(b2), (b3) and (b5) that the method of [11] (block-based method) with the extracted feature from the unified block is unable to detect the large scaling transform. It is shown in Fig. 10-(c1)~(c5) that the method in [18] with a large segmentation easily ignores or misses the small-region forgeries. Therefore, The detection results from the method [18] can detect scaling transform forgeries, but its detection performance for small-region forgeries is weak.

### 4.3 Detection results under plain copy-move and authentic images

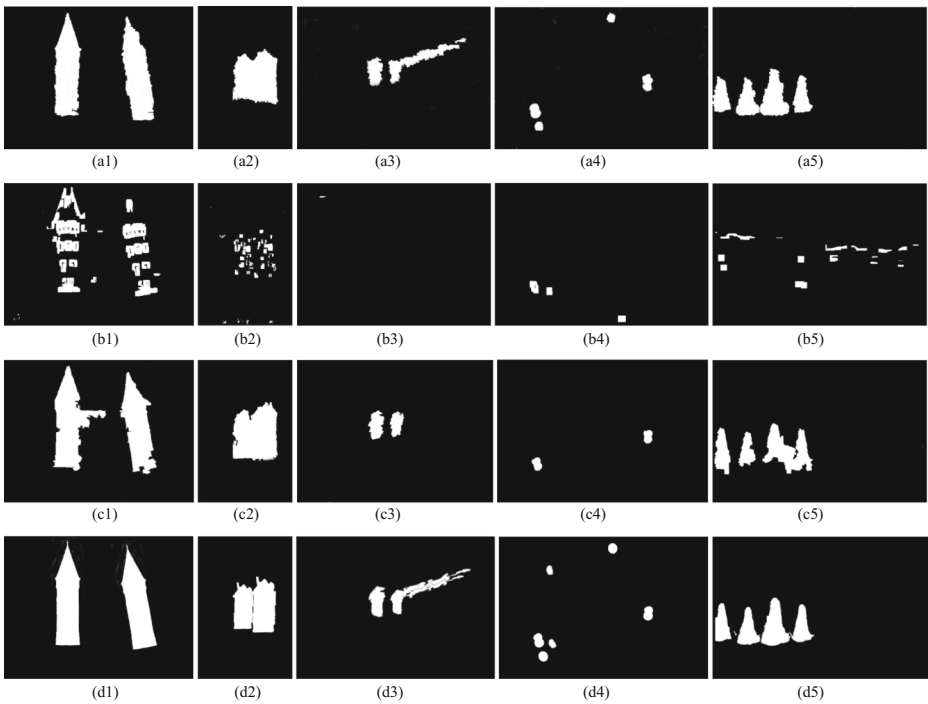
In this sub-section, the experimental results present a comparison of the performance of the proposed method with those of state-of-the-art methods at both the image level and the pixel level. The *precision*, *recall* and  $F_1$  scores are employed to evaluate the plain copy-move forgeries and the authentic image. These experiments were based on the *orig* and *null* sub-datasets. The *orig* sub-dataset contains authentic images with no copy-move operations. The *null* sub-dataset contains copied regions attacked by translation operations. The PCET method [11], SIFT [1, 18, 23] and SURF [27] method results are also provided to evaluate their performances quantitatively. Tables 1 and 2 show the detection results of *precision*, *recall* and  $F_1$  for the CMFD at the image and pixel levels, respectively.





**Fig. 9** The process of the proposed method for CMFD. The first row shows the copy-move forged images. The second row shows the matching results of the candidate keypoint pairs. The third row shows the results of the adaptive keypoint filtering. The fourth through the sixth rows show the 1st, 3rd and ultimate iteration results of suspected regions, respectively. The seventh row shows the relation between the ultimate suspected regions and superpixels

Table 1 shows the detection results of the authentic image and plain copy-move image at the image level. As listed in Table 1, the proposed method achieves relatively high *precision*,



**Fig. 10** The CMFD results of the proposed method. The first to third rows show the detected forgery results of the proposed method, the method from [11] and the method from [18], respectively. The fourth row shows the ground-truth regions

*recall* and  $F_1$ . Our proposed method achieved a *precision* of 96.9%, a *recall* of 93.8% and an  $F_1$  score of 95.3% at the image level. The *precision* of our proposed method was the best compared to the state-of-the-art methods, while the methods from [23, 27] tied for second place. The *precision* of the other methods all exceeded 90%. The *recall* of the methods in [5, 23] achieved the best performance; however, our proposed method is a bit lower than methods in [5, 23]. The  $F_1$  score of our proposed method was only slightly below that of the method from [23]. It was due to the proposed method may abandon some matched pairs, which is an isolated pair or fewer than 3 pairs. So the proposed method may miss some small-sized forgery regions and lower the *recall* score. Table 2 shows the detection results based on the same datasets described in Table 1 and our proposed method achieved the best *recall* score. It is due to the adaptive keypoint filtering procedure which corrects the misclassified keypoints and

**Table 1** Detection results of the plain copy-move and authentic image in image level

Methods	<i>Precision</i> (%)	<i>Recall</i> (%)	$F_1$ (%)
Bi [5]	88.9	100	94.1
Emam [11]	92.7	87.5	90.0
Li [18]	92.1	84.9	88.4
Amerini [1]	88.4	79.2	83.5
Pun [23]	95.8	100	97.9
Shivakumar [27]	95.8	91.7	93.7
Proposed method	96.9	93.8	95.3

**Table 2** Pixel-level detection results of the plain copy-move forgery

Methods	Precision (%)	Recall (%)	$F_1$ (%)
Bi [5]	87.2	89.7	88.4
Emam [11]	88.5	77.1	82.4
Li [18]	83.1	56.3	67.1
Amerini [1]	60.8	71.5	65.7
Pun [23]	96.9	83.3	89.6
Shivakumar [27]	68.7	76.1	72.2
Proposed method	93.8	87.5	90.5

then sharply reduces the classification error of keypoints. It can obtain as many inliers as possible to get the accurate affine matrices for the accurate regions matting. The proposed method achieves a *precision* of 93.8% and the best  $F_1$  score of 90.5% at the pixel level. The  $F_1$  score reflects the overall quality and performance of a CMFD method. The method from [23] captured the highest *precision* and the second-best performance for *recall* and  $F_1$ . Analysis of the above experiment was performed at both the image and pixel-level, the proposed method achieved the best performance at the pixel level and high quality at the image level.

#### 4.4 Detection results under various post-processing conditions

Image-level detection is conducted to automatically detect copy-move forged images, while pixel-level detection is employed to measure the quality a CMFD achieves when detecting the copy-move regions. Therefore, performance at the pixel level is mainly employed to evaluate the performance of CMFD methods. To quantitatively evaluate the performance of the proposed method and the state-of-the-art methods, the measures *precision*, *recall* and  $F_1$  were employed to evaluate the algorithms' performances on down-sampled images with, rotation transforms, scaled transforms and JPEG compression operations at the pixel level.

##### 1) Detection results of down-sampling

These experiments were based on the *nul\_sd*, *scale\_sd*, and *rot\_sd* sub-datasets. The copied regions were attacked only by translation (plain) or rotation or scaling distortions. The scaling factors employed in the *scale\_sd* sub-dataset are 91%, 95%, 99%, 101%, 105% and 109%. The rotation factors employed in the *rot\_sd* sub-dataset are 2°, 4°, 6°, 8° and, 10°. The host images in the sub-datasets down-sampled to 50% of the size of the original images. There were  $48 \times 12 = 576$  tested images. Table 3 shows the down-sampling detection results of *precision*, *recall* and,  $F_1$  for the CMFD methods at the pixel level.

**Table 3** Detection results of down-sample

Methods	Precision (%)	Recall (%)	$F_1$ (%)
Bi [5]	83.8	88.0	85.8
Emam [11]	81.3	72.7	76.6
Li [18]	80.2	52.6	63.5
Amerini [1]	/	/	/
Pun [23]	83.6	73.5	78.2
Shivakumar [27]	56.5	62.5	59.3
Proposed method	84.4	79.2	81.7

## 2) Detection results of rotation transform

These experiments were based on the *rot*, *rotExtra* and *rotExtra2* sub-datasets. The copied regions are attacked by rotation distortions. The attack angles are rotated by 2°, 4°, 6°, 8°, 10°, 20°, 60°, and 180°. There were  $48 \times 8 = 384$  tested images in total. Figure 11 shows the detection results of the CMFD methods against rotation transforms.

## 3) Detection results of scaling transform

These experiments were based on the *scale* and *scaleExtra* sub-datasets. The copied regions are attacked by scaling distortions, and the attacked regions are scaled by 80%, 91%, 93%, 95%, 97%, 99%, 101%, 103%, 105%, 107%, 109% and 120%. There were  $48 \times 12 = 576$  tested images in total. Figure 12 shows the scaling detection results of the CMFD methods against scaling transforms.

## 4) Detection results of JPEG compression

These experiments were based on the *jpeg\_sd* sub-dataset. The copied regions are attacked by the JPEG compression distortion. The quality factor of the forged images reflects compression levels between 20% and 100% with a step size of 10%. There were  $48 \times 9 = 432$  tested images in total. The copied regions are attacked by translation distortions. Figure 13 shows the detection results of the CMFD methods against JPEG compression.

Table 3 shows the detection results on down-sampled images at the pixel level. The ‘/’ means the result is not available. Compared to the other methods, our proposed method achieved the best *precision* of 84.4% and took second place in *recall* and  $F_1$  scores. The multi-level dense descriptor method [5] achieved the best *recall* and  $F_1$ , but at the expense of running times. However, in the following experiments under various geometric distortions and post-processed operations, the performances of method [5] were weaker than the proposed method in pixel level. The SURF method [27] resulted in the weakest performance because it identifies too few keypoints to indicate the ground-truth region. Compared to Table 2, the performance of our proposed method decreases only slightly on down-sampled images. It is because the high matching threshold described in (3) can obtain sufficient keypoints to match the images.

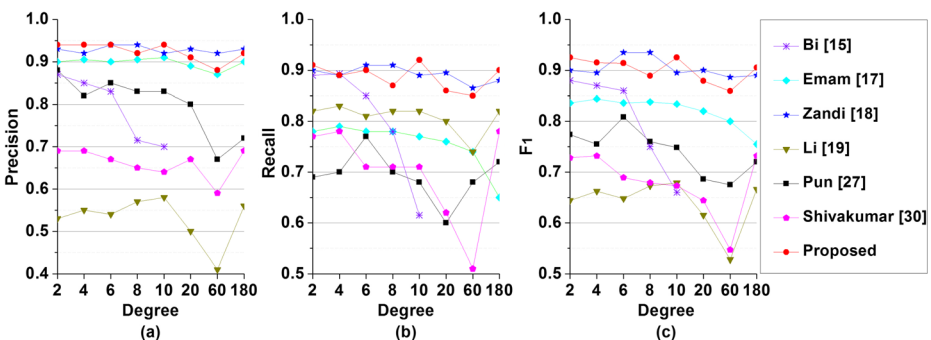
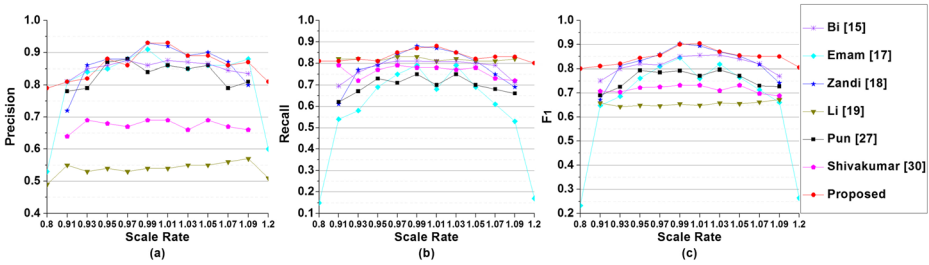
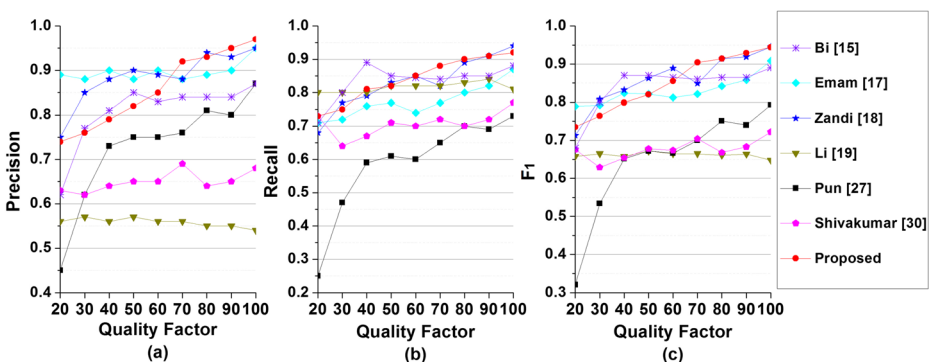


Fig. 11 Detection results of the compared CMFD methods against rotation transform at the pixel level. (a) Precision; (b) Recall; (c)  $F_1$



**Fig. 12** Detection results of the compared CMFD methods against scaling transform at the pixel level. (a) Precision; (b) Recall; (c)  $F_1$

In Figs. 11, 12, and 13, the curve drawn in purple and marked ‘Bi [5]’ represented dense multi-level descriptor of block-based method. The curve drawn in light blue and marked ‘Emam [11]’ represented the results of the block-based PCET method. The curve drawn in dark blue and marked ‘Zandi [32]’ represented the results of the iterative interest-point method. The curves drawn in khaki and black were marked as ‘Li [18]’ and ‘Pun [23]’, respectively, and represented the results of the SIFT methods. The curve drawn in pink and marked ‘Shivakumar [27]’ represented the results of the SURF method. Finally, the curve drawn in red and marked ‘Proposed’ represented the results of the proposed method. The X-axis coordinates represented the rotation degree, scaling factor, and quality factor, respectively. As shown in Fig. 11-(a), the proposed method achieved the best performances in some cases, such as 2°, 4°, 6°, and 10° rotation factors. In other cases, the method from [32] achieved the best performances. The *precision* of our proposed method and the method from [32] both achieved approximately equal performances and achieve the best performance in most of the cases. The method from [11] was slightly weaker than the proposed method and took second place. The proposed method and method [11] both achieved good performances because they used rotation-invariant features. In Fig. 11-(b), the *recall* of our proposed method and that of the method from [32] were much better than those of the other compared methods. The SIFT-based method [18] took third place. In Fig. 11-(c), the  $F_1$  score (which combined both *precision* and *recall*) of both our proposed method and the method from [32] achieved a superior performance compared to the other state-of-the-art methods. It was because these methods fuse the invariance of SURF and the rotation-invariant extraction of PCET. Figure 12



**Fig. 13** Detection results of the compared CMFD methods against JPEG compression at the pixel level. (a) Precision; (b) Recall; (c)  $F_1$

shows the scaling detection ability of the compared method. It can be observed that the *precision*, *recall*, and  $F_1$  score of the proposed method represented the best performance in most cases, especially for large-factor scaling. The *recall* performance greatly exceeded that of the block-based method from [11], the SIFT method from [23] and the other SURF from [27]. Figure 12 also shows that our proposed method was not sensitive under scaling attacks. Its good performance is because the proposed method calculated the affine matrix and then adaptively adjusts the radius of the circular block to resist scaling transforms. The method from [11] performed poorly on the scaling transforms as shown in Fig. 12-(a), (b) and (c); it was not able to address scaling transform especially for large-factor scaling. Figure 13 shows the detected results under JPEG compression attacked as the image quality factor varies. In Fig. 13-(a), the *precision* of our proposed method achieved the best performance when the quality factor is high. When the quality factor is below 70, the *recall* of our proposed method was similar to the *precision* results. Our proposed method achieved its best performance when the quality factor was high. However, even when the quality factor was low, our method took second place only to the method from [32]. As shown in Fig. 13-(c), the  $F_1$  scores of the proposed method were similar to its *precision* and *recall* scores. The  $F_1$ , *Precision*, and *recall* of our proposed method all exceeded 70% against JPEG compression attacks, and it achieved the best performance in most cases. Even when the quality factor is below 60, our method achieved second place. Adaptive keypoint filtering algorithm of the proposed method can filter most of the outliers and get as many inliers as possible. The inliers can accurately locate the copy-move forgery regions. The affine matrices which are obtained from a large number of the inliers, accurately indicate the geometrical transformations of the forgery regions or the geometrical correlations between the pixel pairs. The iterative region algorithm uses the superpixels to complement the region matting. It is a precise region filling algorithm. Therefore, under various post-processing conditions, our method achieves superior performance compared with other state-of-the-art methods.

#### 4.5 The experiments under CMH dataset and other evaluation criteria

There are some other evaluation criteria proposed in the state-of-the-art methods. The evaluation criteria contain True Positive Rate (*TPR*), False Positive Rate (*FPR*) and Accuracy (*ACC*) [28]. *TPR* is the same case to *recall*. *FPR* describes the ratio of the authentic pixels with incorrect detection to all authentic pixels. *ACC* describes the ratio of the pixels with correct detection to all pixels. *ACC* is defined in Eq. (27).

$$ACC = \frac{TPR + (1-FPR)}{2} = \frac{T_P + F_N}{T_P + F_P + T_N + F_N} \quad (27)$$

The Copy-Move Hard (CMH) dataset proposed in [28] has four sub-dataset with total 108 (23 + 25 + 26 + 34) forgery images. *TPR*, *FPR* and *ACC* were proposed to evaluate the performances of the compared methods under CMH dataset. The compared results of the proposed method against other five methods were described in Table 4.

Table 4 shows the detection results under CMH dataset. Compared to the other methods, *FPR* of the proposed method was a little bit weak, but the proposed method achieved the best *ACC* of 90.8% and *TPR* of 83.3%. The method from [28] captured the second-best *ACC* and *TPR*. The proposed method also got the superior performance to the-state-of-the-art methods at *TPR*, *FPR* and *ACC* criteria.

**Table 4** Detection results of CMH dataset

Methods	ACC (%)	TPR (%)	FPR (%)
Popescu [22]	49.5	0.02	0.09
Ryu [24]	65.5	34.3	0.15
Ryu [25]	64.5	34.3	5.35
Xu [27]	82.4	65.8	0.19
Sliva [28]	83.4	71.9	1.22
Proposed method	90.8	83.3	1.63

#### 4.6 Comparison of running times

From the above analysis, the proposed method comprehensively achieved the best performances under various distortions with different datasets. Now, we turn on the attention for an analysis of the computational complexities between the proposed method and the state-of-the-art methods. To improve efficiency and reduce computational times, the proposed method was implemented by mixed-language programming based on MATLAB and C++. We have implemented the method by using MATLAB 2016b, VS2015 and Opencv 3.2 tools. The experiments were performed on a computer with one Intel(R) Xeon(R) E5–2650 @2.20 GHz CPU with 12 cores and 64 GB RAM. We chose the datasets of Friedrich-Alexander University [9] which were larger than the ones in CMH dataset. We have divided the proposed method into three parts which are feature extraction, adaptive keypoint filtering, and iterative region merging stage, respectively. Table 5 shows the average running times of the 3 stages and total times of the proposed method.

There are two difficult problems for complexity comparison between the proposed method and each analyzed method. First, most of the state-of-the-art methods have not provided the running times. Second, each compared method was not performed on a unified platform. It was hard to compare the time of each method exactly. We have tried to provide the running times of the compared methods with the available codes. We have also cited the running times of paper [28] and normalized the running time of the proposed method based on the approximate criteria of the method [28]. Table 6 shows the compared results.

Table 6 clearly shows that the running times of the keypoint-based methods ([7, 23, 28], the proposed method) were generally less than the block-based methods ([5, 11, 22, 25]). The block-based methods took expensive times for feature extraction and matching of each block (or each pixel). The keypoint-based methods extracted local image features as keypoints and only filtered the extracted keypoints as matches. As for the proposed method, the extracted feature stage and keypoint filtering stage occupied a small portion of the total time. It was owing to the mixed-language programming with various effective tools. The iterative region needed to iteratively calculate each neighboring keypoints feature. So the iterative region merging took the relatively expensive time.

**Table 5** Running time of the proposed method using dataset [9]

Stage or the total method	Running times (s)
Feature extraction	6.49
Adaptive keypoint filtering	13.54
Iterative region merging	50.56
Total method	70.59

**Table 6** Running times of the proposed method and the state-of-the-art methods

Methods	Running times (s)
Popescu [22]	180.11
Ryu [24]	56.4
Ryu [25]	1418.68
Bi [5]	720.51
Emam [11]	3851.22
Pun [23]	136.82
Xu [7]	42.56
Sliva [28]	18.81
Proposed method	64.59

## 5 Conclusions

In this paper, the proposed method fused the advantages of the keypoint-based and block-based-methods to perform CMFD. This novel method mainly involves the local maxima, minima extraction and matching algorithm, an adaptive keypoint filtering algorithm, an iterative region-merging algorithm and so on. First, the local maxima and minima are extracted as candidate keypoint pairs by Speeded Up Robust Features (SURF). The Best-Bin-First search (BBF) is employed to obtain the correlation between each pair of candidate keypoints. The candidate keypoints are then matched as pairs by employing the Nearest Neighbors (2NN) test. The adaptive keypoint filtering algorithm based on random sample consensus (RANSAC) is proposed to filter out the outliers and obtain the inlier (keypoint) pairs. The first sub-stage of the filtering algorithm removes nearest neighbor keypoint pairs. The second sub-stage evaluates the inliers and corrects keypoint misclassifications. The adaptive keypoint filtering can identify both single forgeries and multiple forgeries. Finally, the iterative region-merging algorithm is proposed to obtain the forged regions. Neighboring keypoints (NKs) are generated to obtain additional inliers to cover the forged regions, and DPCET is employed to extract the circular block features corresponding to the NKs and filter out any mismatched NKs. The suspected regions are generated by the circular blocks of the matched NKs. Simple Linear Iterative Clustering (SLIC) is employed to segment the host image into superpixels. The superpixels and suspected regions are merged to fill the detected regions more precisely. Compared with the state-of-the-art methods, a series of experiments demonstrated that our proposed method achieves the best performance for CMFD under various post-processing operations. In future work, the adaptive keypoint filtering and iterative region-merging of our proposed scheme will be applied to wider fields, focusing not only on CMFD but also other types of forgery detection such as splicing and image morphing.

**Acknowledgements** This work was supported in part by the Research Committee of the University of Macau under Grant MYRG2018-00035-FST, and the Science and Technology Development Fund of Macau SAR under Grant 041/2017/A1.

## References

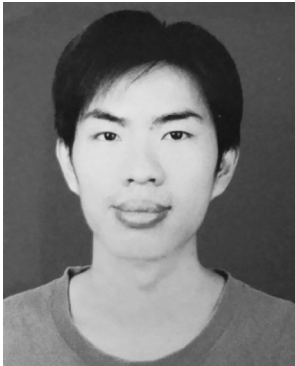
1. Amerini I, Ballan L, Caldelli R, Del Bimbo A, Serra G (2011) A sift-based forensic method for copy–move attack detection and transformation recovery. *IEEE Trans Inform Foren Sec* 6:1099–1110
2. Amerini I, Ballan L, Caldelli R, Del Bimbo A, Del Tongo L, Serra G (2013) Copy-move forgery detection and localization by means of robust clustering with J-linkage. *Signal Process Image Commun* 28:659–669



3. Beis JS, Lowe DG (1997) "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," in *Computer Vision and Pattern Recognition, 1997*. Proc 1997 IEEE Comput Soc Conf:1000–1006
4. Bhullar LK, Budhiraja S, Dhindsa A (2014) DWT and SIFT based passive copy-move forgery detection. *Int J Comput Appl* 95
5. Bi X, Pun C-M, Yuan X-C (2016) Multi-level dense descriptor and hierarchical feature matching for copy-move forgery detection. *Inf Sci* 345:226–242
6. Bi X, Pun C-M, Yuan X-C (2016) Multi-scale feature extraction and adaptive matching for copy-move forgery detection. *Multimed Tools Appl*:1–23
7. X Bo, W Junwen, L Guangjie, D Yuewei (2010) "Image copy-move forgery detection based on SURF," in *Multimedia information networking and security (MINES), 2010 international conference on*. 889–892.
8. R Caldelli, I Amerini, L Ballan, G Serra, M Barni, A Costanzo (2012) "On the effectiveness of local warping against SIFT-based copy-move detection," in *Communications control and signal processing (ISCCSP), 2012 5th international symposium on*. 1–5.
9. Christlein V, Riess C, Jordan J, Riess C, Angelopoulou E (2012) An evaluation of popular copy-move forgery detection approaches. *IEEE Trans Inform Forens Sec* 7:1841–1854
10. S Debbarma, AB Singh, KM Singh (2014) "Keypoints based copy-move forgery detection of digital images," in *Informatics, Electronics & Vision (ICIEV), 2014 international conference on*. 1–5.
11. Emam M, Han Q, Niu XM (2016) PCET based copy-move forgery detection in images under geometric transforms. *Multimed Tools Appl* 75:11513–11527
12. Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun ACM* 24:381–395
13. AJ Fridrich, BD Soukal, AJ Lukáš (2003) "Detection of copy-move forgery in digital images," in *Proceedings of Digital Forensic Research Workshop*.
14. Gan Y, Zhong J (2015) Research on copy-move image forgery detection using features of discrete polar complex exponential transform. *Int J Bifurc Chaos* 25:1540018
15. H Huang, W Guo, Y Zhang (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*. 272–276.
16. Lee J-C, Chang C-P, Chen W-K (2015) Detection of copy-move image forgery using histogram of orientated gradients. *Inf Sci* 321:250–262
17. Li Y (2013) Image copy-move forgery detection based on polar cosine transform and approximate nearest neighbor searching. *Forensic Sci Int* 224:59–67
18. Li J, Li X, Yang B, Sun X (2015) Segmentation-based image copy-move forgery detection scheme. *IEEE Trans Inform Foren Sec* 10:507–518
19. Lin Z, He J, Tang X, Tang C-K (2009) Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis. *Pattern Recogn* 42:2492–2501
20. Liu Q (2017) An approach to detecting JPEG down-recompression and seam carving forgery under recompression anti-forensics. *Pattern Recogn* 65:35–46
21. AM Moussa (2015) "A fast and accurate algorithm for copy-move forgery detection," in *Computer Engineering & Systems (ICCES), 2015 tenth international conference on*. 281–285.
22. A Popescu, H Farid (2004) "Exposing digital forgeries by detecting duplicated image region [technical report]. 2004-515," *Hanover, Department of Computer Science, Dartmouth College. USA*, p. 32.
23. Pun C-M, Yuan X-C, Bi X-L (2015) Image forgery detection using adaptive oversegmentation and feature point matching. *IEEE Trans Inform Foren Sec* 10:1705–1716
24. Ryu S-J, Lee M-J, Lee H-K (2010) Detection of copy-rotate-move forgery using zernike moments. *Int Worksh Inform Hiding*:51–65
25. Ryu S-J, Kirchner M, Lee M-J, Lee H-K (2013) Rotation invariant localization of duplicated image regions based on Zernike moments. *IEEE Trans Inform Foren Sec* 8:1355–1370
26. Saleh SQ, Hussain M, Muhammad G, Bebis G (2013) Evaluation of image forgery detection using multi-scale weber local descriptors. *Int Symp Vis Comput*:416–424
27. Shivakumar B, Baboo LDSS (2011) Detection of region duplication forgery in digital images using SURF. *IJCSI Int J Comput Sci Issues* 8
28. Silva E, Carvalho T, Ferreira A, Rocha A (2015) Going deeper into copy-move forgery detection: exploring image telltales via multi-scale analysis and voting processes. *J Vis Commun Image Represent* 29:16–32
29. K Sudhakar, V Sandeep, S Kulkarni (2014) "Speeding-up SIFT based copy move forgery detection using level set approach," in *Advances in electronics, computers and communications (ICAIECC), 2014 international conference on*, pp. 1–6.
30. B Ustubioglu, V Nabryev, G Ulutas, M Ulutas (2015) "Image forgery detection using colour moments," in *Telecommunications and signal processing (TSP), 2015 38th international conference on*, pp. 540–544.

31. Yap P-T, Jiang X, Kot AC (2010) Two-dimensional polar harmonic transforms for invariant image representation. *IEEE Trans Pattern Anal Mach Intell* 32:1259–1270
32. Zandi M, Mahmoudi-Aznavah A, Talebpour A (2016) Iterative copy-move forgery detection based on a new interest point detector. *IEEE Trans Inform Foren Sec* 11:2499–2512
33. F Zhao, R Zhang, H Guo, Y Zhang (2015) "Effective digital image copy-move location algorithm robust to geometric transformations," in *Signal processing, communications and computing (ICSPCC), 2015 IEEE international conference on*, pp. 1–5
34. Zheng J, Liu Y, Ren J, Zhu T, Yan Y, Yang H (2016) Fusion of block and keypoints based approaches for effective copy-move image forgery detection. *Multidim Syst Sign Process* 27:989–1005
35. Zhong J, Gan Y, Young J, Huang L, Lin P (2017) A new block-based method for copy move forgery detection under image geometric transforms. *Multimed Tools Appl* 76:14887–14903

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Junliu Zhong** received his M.S. degree in Detection technology and automation devices from Sun Yat-sen University in 2006, China. He is currently pursuing the Ph.D. degree in software engineering with the University of Macau. His current research interests include information system security, watermarking, pattern recognition, signal and image processing.



**Chi-Man 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 at 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 International Conference Computer Graphics, Imaging and Visualization (CGIV2013), 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.