Multimedia Tools and Applications (2022) 81:37847–37872
https://doi.org/10.1007/s11042-022-12755-w

Copy-move image forgery detection based on evolving circular domains coverage

Shilin Lu[1,2] · Xinghong Hu[1] · Chengyou Wang[1] · Lu Chen[1] · Shulu Han[1] · Yuejia Han[1]

Received: 6 April 2021 / Revised: 21 December 2021 / Accepted: 21 February 2022 /
Published online: 22 April 2022

Abstract

The aim of this paper is to improve the accuracy of copy-move forgery detection (CMFD) in image forensics by proposing a novel scheme and the main contribution is evolving circular domains coverage (ECDC) algorithm. The proposed scheme integrates both block-based and keypoint-based forgery detection methods. Firstly, the speed-up robust feature (SURF) in log-polar space and the scale invariant feature transform (SIFT) are extracted from an entire image. Secondly, generalized 2 nearest neighbor (g2NN) is employed to get massive matched pairs. Then, random sample consensus (RANSAC) algorithm is employed to filter out mismatched pairs, thus allowing rough localization of counterfeit areas. To present these forgery areas more accurately, we propose the efficient and accurate ECDC algorithm to present them. This algorithm can find satisfactory threshold areas by extracting block features from jointly evolving circular domains, which are centered on matched pairs. Finally, morphological operation is applied to refine the detected forgery areas. Experimental results indicate that the proposed CMFD scheme can achieve better detection performance under various attacks compared with other state-of-the-art CMFD schemes.

Keywords Image forensics · Copy-move forgery detection (CMFD) · Scale invariant feature transform (SIFT) · Speed-up robust feature (SURF) · Evolving circular domains coverage (ECDC)

1 Introduction

With the development of computer and image processing software, digital image tampering becomes much easier; therefore, lots of digital images lack authenticity and integrity, which poses a threat to many critical fields. For example, it may lead to misdiagnosis when forged images are used in medical fields [31], and forged newspaper photographs may mislead

✉ Chengyou Wang
wangchengyou@sdu.edu.cn

[1] School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai 264209, China

[2] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore

people and cause unnecessary social unrest [26]. Hence, the ability to credibly authenticate an image has become a major focus of image forensics and security.

The existing detection techniques fall into two main categories: active and passive. Active forensics techniques ensure the authenticity of digital images by verifying the integrity of authentication information, such as digital watermark [32, 40, 45] and digital signature [21, 22, 38]. These active methods have strong detection abilities and cannot be easily avoided, but their main defect is that the watermark must be inserted as a key into the image. Passive forensics techniques are used to verify the authenticity by analyzing the information and structure of the image, which overcomes the disadvantage of active forensics techniques.

There are two main forgeries that alter the contents of images: splicing and copy-move. The common splicing forgery method consists in copying and pasting a part of an image into another image, while the copy-move forgery method is a way to copy and paste a part of an image into the same image. In recent years, copy-move forgery has become one of the most popular subtopics in forgery detection [7]. To make copy-move tampered images more trustworthy, some processing methods are probably required, including rotation, scaling, downsampling, JPEG compression, and noise addition. Considering that image copy-move forgery detection (CMFD) is a challenging topic, this paper focuses on CMFD algorithms.

The general steps of CMFD are feature extraction, feature matching, and postprocessing. Based on different extracting features, CMFD is divided into block-based, keypoint-based, and fusion of these two methods. The last one has become more popular in recent years.

In this paper, we propose a CMFD scheme based on evolving circular domains coverage (ECDC), which combines block-based and keypoint-based methods. It extracts two different descriptors from an image, and then we match and filter those descriptors to obtain a rough localization. After that, we employ the proposed ECDC algorithm to cover forgery areas. The refined forgery areas are obtained by postprocessing ultimately. The two main contributions of this paper are listed below:

1) we combine the speed-up robust feature (SURF) in log-polar space and the scale invariant feature transform (SIFT) as descriptors to depict a host image more accurately. This not only raises the precision of the proposed scheme under plain copy-move forgery evidently, but also improves its robustness to various geometric transformations and signal processing.
2) we propose a novel algorithm, named ECDC, to present forgery areas exactly. By comparing the differences of block features in jointly evolving circular domains, this algorithm which is based on the pre-positioning of keypoints can greatly reduce computational complexity and improve its running efficiency. In addition, ECDC cannot only cover large-scale tampered areas completely, but also depict small areas accurately.

The rest of this paper is organized as follows: Section 2 briefly reviews the related work of CMFD; Section 3 displays the framework of the proposed CMFD scheme and then explains each step in detail; Section 4 shows the experimental results of CMFD and their analysis; finally, Section 5 gives the conclusion.

2 Related work

In this section, we review some classic and state-of-the-art CMFD schemes. Based on the difference of extracted features from the image, we divide this review into the following parts: block-based method, keypoint-based method, and fusion of the two methods.

2.1 Block-based methods

The block-based CMFD methods, in general, divide a host image into small, regular, and overlapped blocks. After extracting features of each subblock, the results are obtained by matching and postprocessing those features. Fridrich et al. [12] proposed the CMFD algorithm, which is a milestone in the field of CMFD. They used quantified discrete cosine transform (DCT) coefficients as features. Then, a lexicographically ordered feature matrix reducing the range of feature matching, was used to detect similar regions [7]. Popescu and Farid [27] applied principal components analysis (PCA) to generate a reduced dimension representation so as to improve the robustness of the detection. Bayram et al. [4] proposed Fourier-Mellin transform (FMT) to extract features. They applied counting bloom filters instead of Lexicographic sorting as a matching scheme which was more efficient. Wang et al. [36, 37] used the Gaussian pyramid to reduce the dimensions of images. The former used the Hu-moments of blocks, and the latter employed the mean value of image pixels in circle blocks, which were divided into concentric circles. Ryu et al. [30] proposed a method based on rotationally-invariant Zernike moments, which can detect forged regions even though they are rotated. Li [16] proposed an algorithm that matched polar cosine transform (PCT) with locality sensitive hashing (LSH), which required simpler calculations than Zernike moments. This algorithm excels at large-scale rotation. Similarly, polar sine transform (PST) and polar complex exponential transform (PCET) also belong to polar harmonic transform (PHT) [43]. Bravo-Solorio and Nandi [5] used colour-dependent feature vectors to perform an efficient search in terms of memory usage. Cozzolino et al. [8, 9] proposed a new matching method called PatchMatch, and a fast postprocessing procedure based on dense linear fitting. This method greatly reduces the computational complexity and it is robust to various types of distortions.

Overall, although applying Lexicographic sorting and reducing dimensions make block-based methods detection more efficient, it still has higher computational complexity than keypoint-based methods. In addition, when faced with large-scale scaling, the robustness of block-based methods, in general, is significantly reduced.

2.2 Keypoint-based methods

The keypoint-based CMFD methods usually extract features from an entire image, which is the main difference from block-based methods, and they effectively reduce computational complexity. Huang et al. [14] proposed the best-bin-first nearest neighbor identification algorithm based on SIFT. Xu et al. [42] proposed SURF to extract features with a faster speed compared with SIFT. Amerini et al. [1] used generalized 2 nearest neighbor (g2NN) on SIFT descriptor to obtain qualified features. Then the random sample consensus (RANSAC) was used to remove mismatched points. Shivakumar and Baboo [33] proposed a CMFD scheme based on SURF and kd-tree for multidimensional data matching. In high-resolution image processing process, this method can detect different sizes of copied regions with a minimum number of false matches. To present tampered areas accurately, Pan and Lyu [23] utilized RANSAC to estimate the affine transformation matrix, and then they obtained correlation maps by calculating correlation coefficients to locate forged regions. Silva et al. [34] proposed to separate forged points and the corresponding original ones into different clusters by clustering matched keypoints based on their locations and the final decision is based on a voting process. Park et al. [25] utilized SIFT and the reduced local binary pattern (LBP) histogram to detect tampered areas.

However, [1, 33] only roughly marked the detected regions with connections on matched pairs. Furthermore, when tampering occurs in low-entropy or small-size areas, the detection results of many keypoint-based methods are unsatisfying due to the small number of keypoints.

2.3 Fusion of block-based and keypoint-based methods

For better detection performance, combining the advantages of block-based and keypoint-based methods have currently become a trend. Some researchers proposed to segment the host image into non-overlapped and irregular blocks and then to match features extracted from those segmented regions [17, 29]. But their accuracy depends on the size of superpixels and detected results may have fuzzy boundaries. Zheng et al. [46] classified the host image into textured and smooth regions in which, SIFT and Zernike features were respectively extracted and matched. However, this method cannot accurately distinguish between smooth and textured areas, especially when tampered regions are attacked by noise. Zandi et al. [44] proposed a new interest point detector and used an effective filtering algorithm and an iteration algorithm to improve their performance. Although they can effectively detect tampered areas in low contrast areas, their detected results usually contain mismatches. Pun and Chung [28] proposed a two-stage localization for CMFD. The weber local descriptor (WLD) was extracted from each superpixel in their rough localization stage, and in their precise localization stage, discrete analytic Fourier-Mellin transform (DAFMT) of roughly located areas were extracted. Li and Zhou [18] proposed a hierarchical matching strategy to improve the keypoint matching problems and an iterative localization technique to localize the forged areas. Wang et al. [39] classified irregular and non-overlapping image blocks into smooth and textured regions. They combined RANSAC algorithm with a filtering strategy to eliminate false matches. This method can detect a high-brightness smooth forgery. However, these methods achieve high detecting accuracy at the expense of low efficiency.

In summary, the main problems faced by block-based CMFD methods are the inability to detect images with large-scale scaling and high computational complexity, while the main problem of keypoint-based CMFD methods is that there are fewer keypoints in low-entropy areas, which lead to incomplete coverage of tampered areas. Fusing block-based and keypoint-based methods reasonably can preserve their advantages and avoid certain shortcomings at the same time. Our scheme fairly integrates block-based and keypoint-based methods, which results in complete coverage of tampered areas and higher detection efficiency. The algorithm is described in more detail in Section 3.

3 Proposed copy-move forgery detection scheme

In this section, we explicate our CMFD scheme. The framework of the whole scheme is given in Fig. 1. Firstly, we extract both SIFT descriptor and log-polar SURF descriptor (LPSD) from an entire image. Secondly, g2NN is employed on each descriptor to obtain massive matched pairs. Then, we employ RANSAC to eliminate mismatched pairs. Finally, the ECDC algorithm is used to present entire forgery regions through those matched pairs. In the rest of this section, Section 3.1 explains the feature extraction algorithm combining SIFT and LPSD; Section 3.2 introduces the keypoints matching algorithm using g2NN; Section 3.3 describes the process of eliminating mismatched pairs by using RANSAC; Section 3.4 explains in detail how matched pairs are expanded to whole forgery regions by using ECDC algorithm.

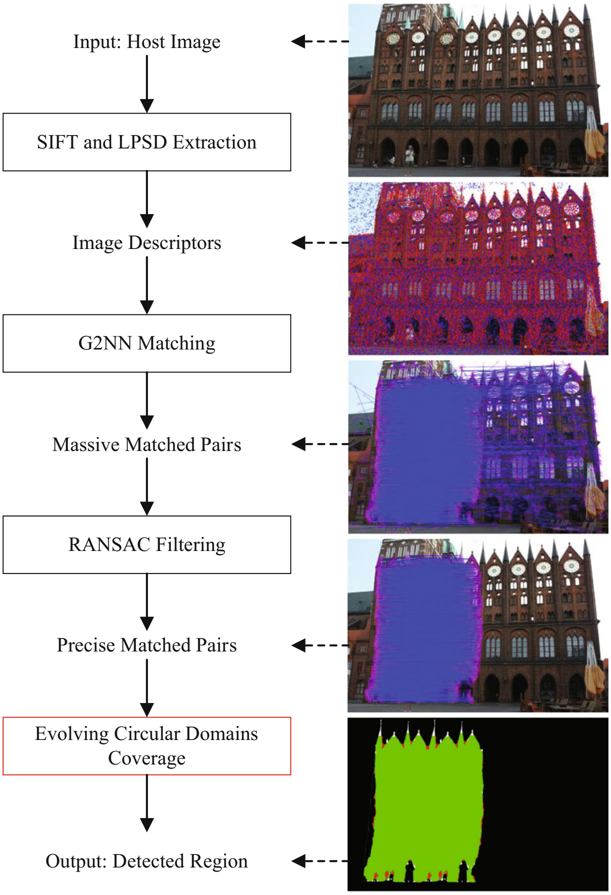


Fig. 1 Framework of the proposed copy-move forgery detection scheme. In the second image, SIFT is labeled with blue circles and LPSD is labeled with red dots

3.1 Feature extraction using combination of SIFT and LPSD

In this section, we explain how to extract keypoints as descriptors of the image. SIFT and SURF algorithms have been widely used in the field of computer vision in recent years. These keypoints are robust to various attacks, including rotation, scaling, downsampling, JPEG compression, and noise addition. As a result, SIFT and SURF are often used to extract keypoints in existing keypoint-based methods. In this paper, unlike in general keypoint-based methods, we combine SIFT and LPSD to depict images.

3.1.1 SIFT

Lowe [19] decomposed the SIFT algorithm into the following four steps: firstly, extrema in scale space were located with the computation searching over all scales and image locations; secondly, at each candidate location, keypoints were selected based on measures of their stability; then, based on local image gradient directions, one or more orientations were assigned to each keypoint location; at last, the local image gradients were measured at the selected scale in the region around keypoint to generate descriptors.

In general, the extreme points of a given image are detected at different scales in scale space, which is constructed by using the Gaussian pyramids with different Gaussian smoothing and resolution subsampling. These keypoints are extracted by applying difference of Gaussian (DoG), and a DoG image D is denoted by [19]:

$$\begin{aligned} D(x, y, \sigma) &= [G(x, y, k\sigma) - G(x, y, \sigma)] * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma), \end{aligned} \tag{1}$$

where $L(x, y, k\sigma)$ is the convolution of the original image $I(x, y)$, with the Gaussian blur $G(x, y, \sigma)$ at scale space k.

To ensure rotation invariance, for each keypoint, the algorithm assigns a canonical orientation which can be determined by calculating the gradient in its neighborhood. Specifically, for an image sample $L(x, y, \sigma)$ at scale σ , the gradient magnitude $m(x, y)$ and orientation $\theta(x, y)$ can be pre-calculated using pixel difference as follows [19]:

$$\begin{aligned} m(x, y) = \; & [[L(x+1, y) - L(x-1, y)]^2 \\ & + [L(x, y+1) - L(x, y-1)]^2]^{\frac{1}{2}}, \end{aligned} \tag{2}$$

$$\theta(x, y) = \tan^{-1} \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}. \tag{3}$$

3.1.2 SURF

SURF proposed by Bay et al. [3] is an improvement on SIFT, and being faster is its prominent characteristic. By using a Hessian matrix for optimization, SURF algorithm accelerates SIFT detection process without reducing the quality of the detected points. Then, box filters of different size are used to establish scale space and to convolute with the integral image. Given a point $\boldsymbol{x} = (x, y)$ in an image $\boldsymbol{I}$, the Hessian matrix $\boldsymbol{H}(\boldsymbol{x}, \sigma)$ in $\boldsymbol{x}$ at scale σ is represented as follows [3]:

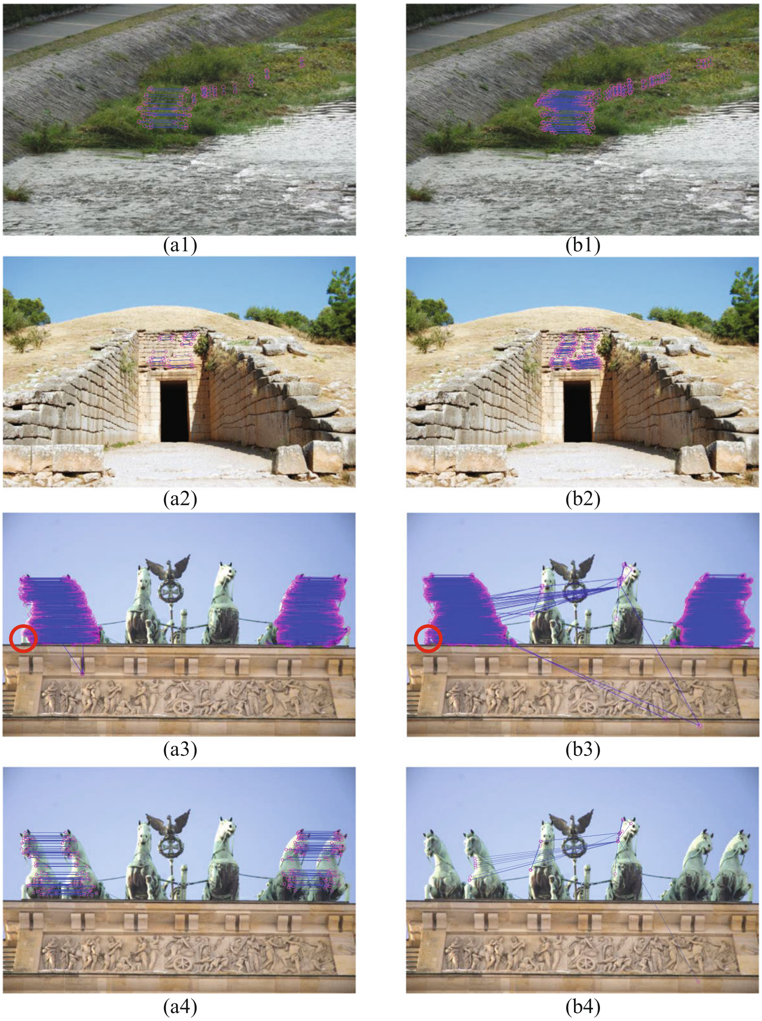
$$\boldsymbol{H}(\boldsymbol{x}, \sigma) = \begin{bmatrix} L_{xx}(\boldsymbol{x}, \sigma) & L_{xy}(\boldsymbol{x}, \sigma) \\ L_{xy}(\boldsymbol{x}, \sigma) & L_{yy}(\boldsymbol{x}, \sigma) \end{bmatrix}, \tag{4}$$

where $L_{xx}(\boldsymbol{x}, \sigma)$ is the convolution result of the second order derivative of Gaussian filter with the image $\boldsymbol{I}$ in point $\boldsymbol{x}$, and similarly for $L_{xy}(\boldsymbol{x}, \sigma)$ and $L_{yy}(\boldsymbol{x}, \sigma)$.

Hessian matrix and non-maximum suppression are used to detect potential keypoints. While assigning one or more canonical orientations, the dominant orientation of the Gaussian weighted Harr wavelet responses can be detected by a sliding orientation window at every sample point within a circular neighborhood around the interest point.

3.1.3 Combination of SIFT and LPSD

Kaura and Dhavale [15] showed that the combination of SIFT and SURF would improve the detection performance of the keypoint-based method. In consideration of the lower detection accuracy of SURF, compared with SIFT [24], we improve this accuracy by applying log-polar transformation to it [35]. It can be seen in experiments that SURF in log-polar space, whose detection results are much more accurate than SIFT, succeeds well in detecting plain copy-move forgery, especially for detailed objects. Figure 2a1–a3 and b1–b3 show SIFT and LPSD matched results for plain copy-move forgery, respectively. (The matching algorithm is explained in Section 3.2). We can observe that LPSD can obtain more matched pairs on small or detailed areas from Fig. 2a3 and b3. However, SIFT exhibits a surprising stability when forgery regions are attacked by noise or any other manipulations, as shown in Fig. 2a4 and b4. In these two figures, noise with standard deviation of 0.1 has been added to the copied fragments. In this case, LPSD hardly detects any right matched pairs while SIFT performs well. Thus, we decide to combine SIFT and LPSD to improve the instability of LPSD and the accuracy of SIFT.


(a1) (b1)

(a2) (b2)

(a3) (b3)

(a4) (b4)

Fig. 2 Comparison of SIFT (left) and LPSD (right) detection results. First three rows: SIFT and LPSD detection results under plain copy-move forgery; fourth row: SIFT and LPSD detection results under local noise attack where the standard deviation is 0.1

3.2 Multiple keypoints matching

3.2.1 g2NN

After feature extraction, two descriptor groups are obtained:

$$\boldsymbol{f}^{\text{SIFT}} = \{f_1{}^{\text{SIFT}}, f_2{}^{\text{SIFT}}, \cdots, f_n{}^{\text{SIFT}}\}, \tag{5}$$

$$\boldsymbol{f}^{\text{LPSD}} = \{f_1{}^{\text{LPSD}}, f_2{}^{\text{LPSD}}, \cdots, f_m{}^{\text{LPSD}}\}, \tag{6}$$

where $\boldsymbol{f}^{\text{SIFT}}$ is the n dimensional SIFT descriptor vector and $\boldsymbol{f}^{\text{LPSD}}$ is the m dimensional LPSD descriptor vector. To find similar descriptors in the image, we need to match them to each other. Lowe [20] employed the distance ratio between the nearest neighbor and the second-nearest neighbor to compare it with a threshold T. Only if the ratio is less than T, the keypoints are matched. However, this matching process is unable to manage multiple keypoints matching. Since the same image areas may be cloned over and over in a tampered image, we employ g2NN algorithm [1] which can cope with multiple copies of the same descriptors. Specifically, taking SIFT as an example, we define a sorted distance vector $\boldsymbol{\chi}_i$ for $\boldsymbol{f}_i^{\text{SIFT}}$ to represent the Euclidean distance between $\boldsymbol{f}_i^{\text{SIFT}}$ and the other $(n-1)$ descriptors, i.e.,

$$\boldsymbol{\chi}_i = \{d_{i,1}, d_{i,2}, \cdots, d_{i,n}\}, \tag{7}$$

where $d_{i,j}$ $(i, j = 1, 2, \cdots, n;\ i \neq j)$ is the Euclidean distance between $\boldsymbol{f}_i^{\text{SIFT}}$ and $\boldsymbol{f}_j^{\text{SIFT}}$, i.e.,

$$d_{i,j} = \left\| f_i^{\text{SIFT}} - f_j^{\text{SIFT}} \right\|_2. \tag{8}$$

To facilitate the finding of an appropriate threshold, we measure the similarity between descriptors by using $d_{i,j}^2$(the Euclidean distance square). Thus, for all $\boldsymbol{f}^{\text{SIFT}}$, an $n \times (n-1)$ matrix $\boldsymbol{\xi}$ is generated:

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\chi}_1^2 \\ \boldsymbol{\chi}_2^2 \\ \vdots \\ \boldsymbol{\chi}_n^2 \end{bmatrix} = \begin{bmatrix} d_{1,2}^2 & d_{1,3}^2 & \cdots & d_{1,n}^2 \\ d_{2,1}^2 & d_{2,3}^2 & \cdots & d_{2,n}^2 \\ \vdots & \vdots & \ddots & \vdots \\ d_{n,1}^2 & d_{n,2}^2 & \cdots & d_{n,n-1}^2 \end{bmatrix}. \tag{9}$$

We iterate 2 nearest neighbor (2NN) algorithm on every row of the distance matrix $\boldsymbol{\xi}$ to find multiple copies. Based on $\boldsymbol{\chi}_i$ as an example, the iteration stops when

$$d_{i,j}^2 / d_{i,j+1}^2 < T, \tag{10}$$

where $T \in (0, 1)$. If the iteration stops at $d_{i,k}^2$, each keypoint corresponding to the distance in $\{d_{i,1}^2, d_{i,2}^2, \cdots, d_{i,k}^2\}$ (where $k = 1, 2, \cdots, n;\ k \neq i$) is considered as a match for the inspected keypoint.

3.2.2 Threshold *T*

Huang et al. [14] analyzed that, if the ratio T of the distance is reduced, then the number of matched keypoints will be reduced, but the matching accuracy will be improved. To test and verify this conclusion, we set different thresholds and observe the number of matched pairs and mismatched pairs of $\boldsymbol{f}^{\text{SIFT}}$ and $\boldsymbol{f}^{\text{LPSD}}$ under plain copy-move forgery. We use Figs. 3 and 4 to perceptibly and statistically describe the result. Figure 3a1–a4 and b1–b4 show separately detected results of SIFT and LPSD, where thresholds range from 0.1 to 0.7 in steps of 0.2. To select an appropriate threshold, we randomly selected 100 images, including

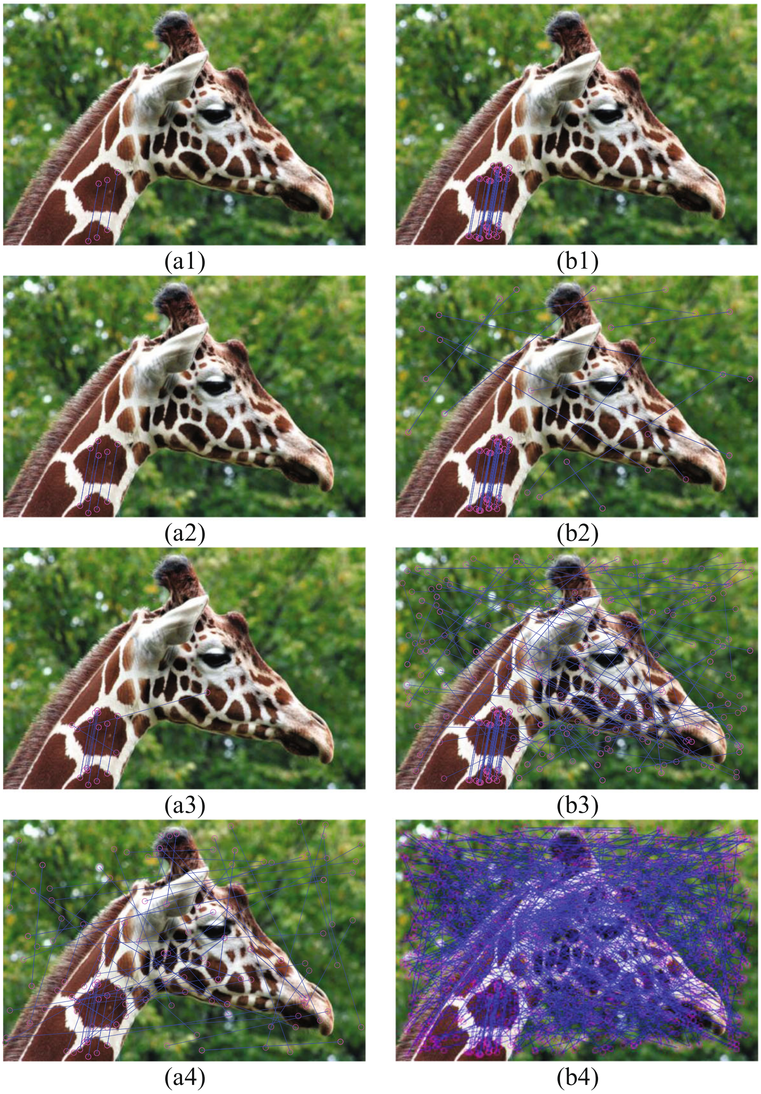
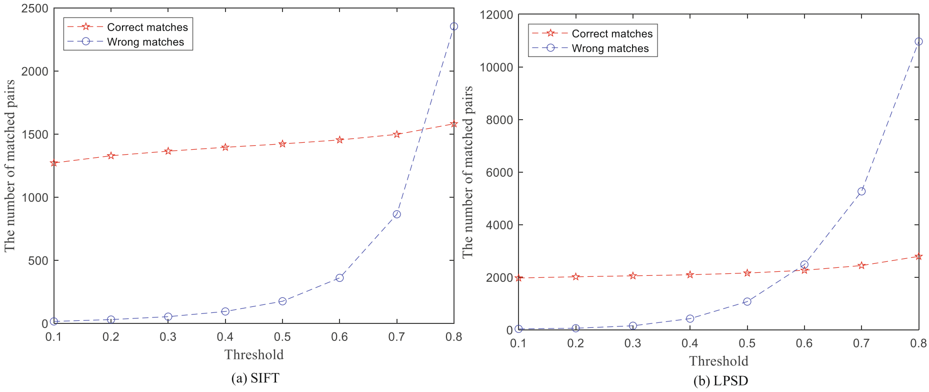


Fig. 3 Comparison of matches under different g2NN thresholds for SIFT (left) and LPSD (right) descriptors

plain copy-move, rotation, scaling, noise, and other attacks, from the FAU dataset [7] for g2NN testing. Statistics data of SIFT and LPSD correct and wrong matches at different thresholds are respectively plotted as a line chart in Fig. 4a and b.

From Fig. 4, we can observe that with the increase of the threshold, correct matches tend to be constant, while incorrect matches increase rapidly. Thus, we come to two conclusions:

1) A higher threshold leads to more false matches, while a lower one may miss some correct matches. It is believed that appropriate threshold should not only obtain as many correct matches as possible, but also guarantee the number of incorrect matches within acceptable limits.



(a) SIFT

(b) LPSD

Fig. 4 The matching results using different g2NN thresholds with (**a**) SIFT and (**b**) LPSD. The correct numbers of matches are depicted in red dashed line with pentagon, while the wrong matches are depicted in dashed line with blue circle

2) Because LPSD has more mismatches at lower thresholds than SIFT descriptor, we set different g2NN thresholds T_{SIFT} and T_{LPSD} for them. The parameters used are presented in Section 4.1.

3.3 Multiple keypoints matching

After keypoints matching, a large number of matched pairs are obtained. Due to the fact that adjacent keypoints have high similarity, we must remove the matched pairs when

$$\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} < S, \tag{11}$$

where (x_a, y_a) and (x_b, y_b) indicate the coordinates of matched keypoints, and S is a predetermined threshold.

However, after that, many mismatched pairs still remain, which seriously puts an negative impact on covering or presenting forgery areas. Thus, we employ a widely used and robust algorithm named RANSAC [11] to eliminate them. RANSAC algorithm can estimate a model parameter precisely even when there are lots of mismatched pairs. It divides those pairs into inlier and outlier groups. To get enough matched pairs and, at the same time, to eliminate mismatched pairs with high similarity, our RANSAC algorithm is based on [39].

We set the threshold N and repeat RANSAC algorithm until the inlier groups points number is less than N. The higher N is, the more mismatched pairs are eliminated. Meanwhile, those slight forgery or low-entropy regions are more likely to be overlooked. By contrast, a lower N is better for detecting those regions. However, it can cause difficulty in eliminating mismatched pairs with high similarity. Therefore, we should get the right balance between the two contradictions.

3.4 Forgery areas coverage algorithm

After postprocessing, we get a number of precisely matched pairs; however, these matched pairs can only cover tampered areas partially, which means that the original appearance of those areas cannot be fully revealed. Hence, accurately covering tampered areas is pivotal in CMFD.

In fact, the matching results of block-based methods and keypoint-based methods are essentially the position of two sets of pixels. More specifically, for the coverage of tampered areas, block-based methods require to compare many image block features centered on pixels. If the features of two blocks are sufficiently similar, their central pixels are recorded as a pair of matched points and their corresponding blocks are subsequently covered. Similarly, we consider that keypoint-based methods can also achieve the goal of covering tampered areas by comparing features, which are within a certain range and centered on pixel points. With the help of keypoint prepositioning, the algorithm complexity can be greatly reduced, thereby improving its detection speed. Thus, we propose a new algorithm to cover tampered areas, which is called ECDC.

3.4.1 Selection of an appropriate feature

Then, for a better coverage, we analyzed and discussed a variety of features. Christlein et al. [7] listed most of the effective features, including four types: moment-based, dimensionality reduction-based, intensity-based, and frequency domain-based features. The DCT coefficients of the frequency domain-based features perform well against noise attacks. Wang et al. [39], through experiments, concluded that PCET moments perform better than other moment-based features under various geometric transformations. Therefore, we chose DCT coefficients and PCET moments for subsequent experiments.

3.4.2 Block feature matching

We extract block features from two separate circular domains centered on a matched pair. Then, we compare those features, and if they are similar enough, the corresponding circular domains are covered. However, different features have different ways to measure their similarity. We usually employ the Euclidean distance to measure the resemblance of PCET moments because its dimension is constant. If the Euclidean distance between $\boldsymbol{F}_1^{\mathrm{PCET}}$ and $\boldsymbol{F}_2^{\mathrm{PCET}}$ is smaller than the predefined threshold K_{PCET}, it is considered as a matched pair, i.e.,

$$\left\| \boldsymbol{F}_1^{\mathrm{PCET}} - \boldsymbol{F}_2^{\mathrm{PCET}} \right\|_2 < K_{\mathrm{PCET}}. \tag{12}$$

Concerning the DCT coefficients, the dimension of the matrices depends on the size of sub image blocks. Consequently, large sub image blocks are stored in grand matrices, which is not conducive to computation. Thus, we use singular value decompositions (SVD) [2, 13] to decompose the extracted DCT coefficients matrices, i.e.,

$$\boldsymbol{F}^{\mathrm{DCT}} = \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{V}^{\mathrm{T}}, \tag{13}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary matrices and $\boldsymbol{\Lambda}$ is a diagonal matrix whose entries are the singular values of $\boldsymbol{F}^{\mathrm{DCT}}$. Since $\boldsymbol{\Lambda}$ contains the basic information of $\boldsymbol{F}^{\mathrm{DCT}}$, and its maximum value includes most of the basic information of $\boldsymbol{F}^{\mathrm{DCT}}$, we choose the maximum value λ of $\boldsymbol{\Lambda}$ to represent $\boldsymbol{F}^{\mathrm{DCT}}$ of a circular domain, i.e.,

$$\lambda = \max(\boldsymbol{\Lambda}). \tag{14}$$

If the difference between λ_1 and λ_2 of two circular domains is less than the threshold K_{DCT}, i.e.,

$$|\lambda_1 - \lambda_2| < K_{\mathrm{DCT}}, \tag{15}$$

we determine that these two circular domains are tampered areas. We take 48 images from the FAU dataset [7], crop their center into sub image blocks of 3 × 3, 39 × 39, and 75 × 75 sizes, and attack them in various ways. Then, we calculate the mean value of λ (denoted as $\bar{\lambda}$) in these three sets of sub image blocks, and list the results in Table 1. It shows that $\bar{\lambda}$ has only a slight difference under various attacks, which proves the feasibility of representing $\boldsymbol{F}^{\mathrm{DCT}}$ by λ to depict sub image blocks.

The selection of the aforementioned thresholds K_{PCET} and K_{DCT} has a great influence on the accuracy and robustness of our algorithm. If they decrease, the criteria get more stringent and the coverage is more precise; however, if the image is attacked by noise and geometric transformations, this algorithm would more easily miss or misjudge tampered areas. On the contrary, it is more robust. These thresholds can be determined through a large number of experiments and they depend on the image resolution and attack type of datasets. For FAU dataset [7], to make it more robust to noise attacks, we set those thresholds as functions $K_{\mathrm{PCET}}(\sigma_{\mathrm{s}})$ and $K_{\mathrm{DCT}}(\sigma_{\mathrm{s}})$, where σ_{s} is the difference of variance between two circular domains. Finally, based on numerous experiments, we have established two empirical formulas for $K_{\mathrm{PCET}}(\sigma_{\mathrm{s}})$ and $K_{\mathrm{DCT}}(\sigma_{\mathrm{s}})$, which are piecewise functions:

$$K_{\mathrm{PCET}}(\sigma_{\mathrm{s}}) = \begin{cases} 1, & \sigma_{\mathrm{s}} \leq 0.1, \\ 25, & 0.1 < \sigma_{\mathrm{s}} \leq 1, \\ 75, & \sigma_{\mathrm{s}} > 1, \end{cases} \tag{16}$$

$$K_{\mathrm{DCT}}(\sigma_{\mathrm{s}}) = \begin{cases} 25, & \sigma_{\mathrm{s}} \leq 1, \\ 50, & 1 < \sigma_{\mathrm{s}} \leq 10, \\ 100, & \sigma_{\mathrm{s}} > 10. \end{cases} \tag{17}$$

3.4.3 Circular domains evolution

Since tampered areas sizes are uncertain, they may not be covered ideally if only the features within a single radius are used as coverage basis. Therefore, we set the radius to an evolving vector in steps of τ:

$$\boldsymbol{R} = \{r_1, r_2, \cdots, r_m\}, \tag{18}$$

where $r_1 < r_2 < \cdots < r_m$. In this way, we can compare the features of matched pairs in an evolving radius range by looping.

The detail of ECDC is illustrated in Fig. 5, in which radii changing process for a keypoint of a matched pair is shown in closeup. For ease of interpretation, the rings in the closeup are labeled with different colors. In the first comparison, we compare the features in the red ring centered on one of the matched pair. When the threshold $K \in \{K_{\mathrm{PCET}}, K_{\mathrm{DCT}}\}$ is met, the radius is enlarged to the size of the blue ring and a second round of comparison is made. If the difference between the features in the blue ring is still less than K, the radius continues to be enlarged until it reaches its maximum or the difference no longer fulfills that condition. Then, the previous radius is recorded and the loop is broken. After traversing all matched pairs with the above algorithm, their coverage is finally completed.

Table 1 Comparison of image blocks $\bar{\lambda}$ under different attacks

Image Types	$\bar{\lambda}(3 \times 3)$	$\bar{\lambda}(39 \times 39)$	$\bar{\lambda}(75 \times 75)$
Original	328.87	3867.53	7454.32
Rotation(15°)	317.85	3866.56	7434.00
Scaling(98%)	328.85	3800.30	7314.76
Noise(0.06)	330.31	3891.77	7496.27

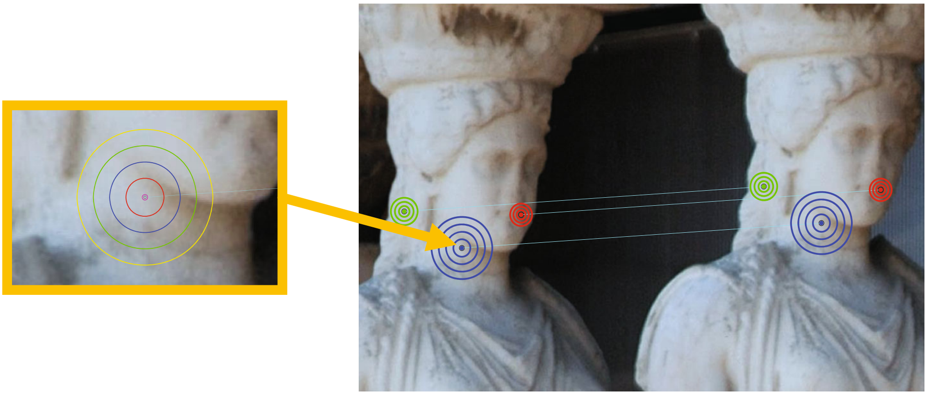


Fig. 5 Illustration of the detail of ECDC algorithm. Left: a locally enlarged radii changing process for a keypoint of a match. Right: the radii expansion process of three sets of matched pairs based on ECDC algorithm. Each matched pair is labeled with different colors

The position of matched pairs is also of great importance on radii expansion. Three expansion results are presented in Fig. 5. The red and green pairs are near the edges of the tampered areas; thus, their rings' extension ends before the radius enlarged to its maximum r_m, which means that ECDC can accurately distinguish the edges. On the contrary, the blue pair is in the center of the tampered areas and, obviously, surrounded by it, so the expansion of the blue ring ends when the radius enlarges to its maximum r_m.

Figure 6 presents the flowchart of ECDC algorithm, in which the middle image only partially shows the coverage of matched pairs. Furthermore, Fig. 5 represents the enlarged and detailed diagram of the step 'Threshold Comparison Repetition' of the loop in Fig. 6.

3.4.4 Morphological postprocessing

Finally, depending on the image resolution, the disk size used for close operation varies. This step fills small holes and cracks in the merged areas while maintaining the overall outline of the areas as it is, which is advantageous to completely cover tampered areas.

4 Experimental results and analysis

In this section, we conducted a series of experiments to compare validity and robustness between our scheme and other state-of-the-art schemes. Section 4.1 presents datasets we used, experimental setup, and parameters. Section 4.2 presents how we evaluated CMFD schemes. Section 4.3 presents the comparison between the proposed and other CMFD schemes at pixel level. Section 4.4 presents the comparison between the proposed and other CMFD schemes at image level.

4.1 Image datasets

For a comprehensive comparison, three datasets, i.e., FAU [7], GRIP [9], and COVERAGE [41] are used to demonstrate the effectiveness of our scheme. FAU [7] dataset consists of 48 high-resolution images and it contains sub-datasets under various image attacks, including

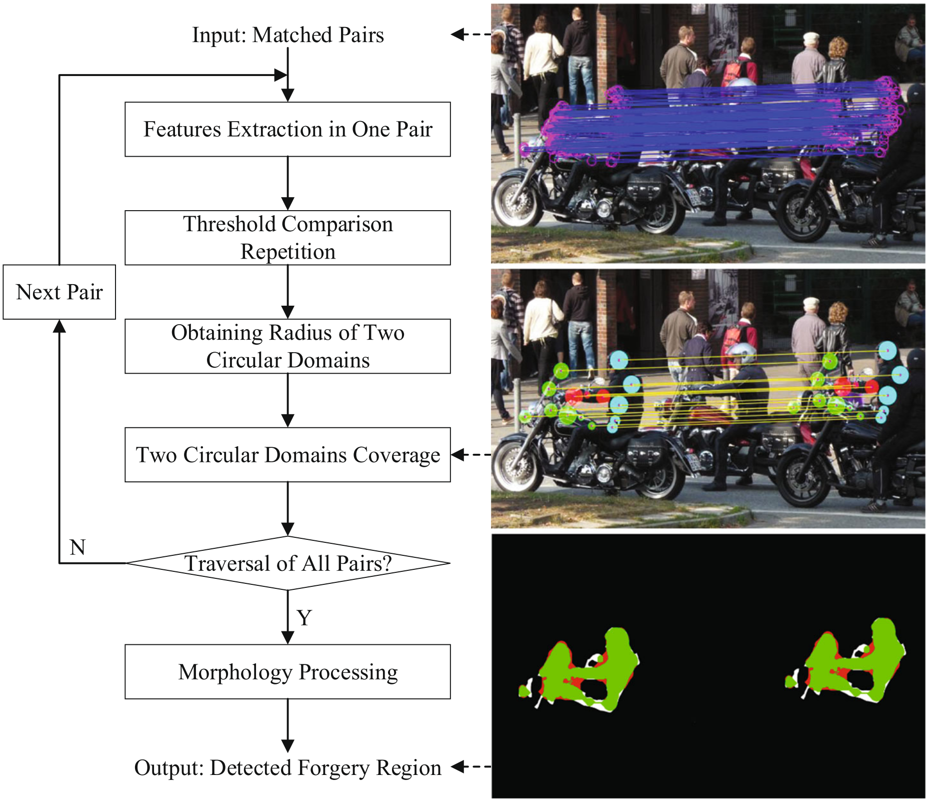

Input: Matched Pairs
Features Extraction in One Pair
Threshold Comparison Repetition
Next Pair
Obtaining Radius of Two Circular Domains
Two Circular Domains Coverage
N
Traversal of All Pairs?
Y
Morphology Processing
Output: Detected Forgery Region

Fig. 6 Flowchart of ECDC algorithm

scaling, rotation, noise, downsampling, and JPEG compression. GRIP [9] only has plain copy-move images but some very smooth tampered areas, while COVERAGE [41] contains similar-but-genuine objects under a combination of different attacks. Hence, we chose FAU [7] to objectively evaluate CMFD schemes at pixel level, and GRIP [9], COVERAGE [41] to evaluate them at image level. The detailed information of these three datasets is summarized in Table 2.

The experiments in this paper were performed in MATLAB 2019b on a 64-bit win10 PC with the Intel Core i7-8650 CPU model and 8 GB RAM. Finally, we listed the parameters used in the proposed scheme in Table 3.

Table 2 Detailed information of datasets

Dataset	Average Resolution	Number of Images	Image Format
FAU [7]	1500 × 1500	Authentic:48 Tampered:1968	PNG
GRIP [9]	1024 × 768	Authentic:80 Tampered:80	PNG
COVERAGE [41]	400 × 486	Authentic:110 Tampered:110	TIF

Table 3 Parameters setting in the proposed scheme

Parameter	Value	Meaning
T_{SIFT}	0.6	Threshold of SIFT in g2NN test
T_{LPSD}	0.1	Threshold of LPSD in g2NN test
S	50	Threshold of Euclidean distance
r_1	1.5	Minimum value of radii group
r_m	37.5	Maximum value of radii group
τ	2	Step of radii group

4.2 Evaluation metrics

Some state-of-the-art schemes uses True Positive Rate (TPR), False Positive Rate (FPR) and Accuracy (ACC) [10, 18] to evaluate their performance, while some choose precision p, recall r [7, 29, 46] and F_1 score. To comprehensively evaluate CMFD methods, these two different metrics are used at two different levels.

At the image level, we focus on the practicality of our scheme to evaluate whether it can distinguish or not the difference between authentic images and forged images, as our original intention is to expose digital image forgery. In this case, metrics TPR, FPR and ACC are used. In CMFD schemes, the TPR t indicates the percentage of correctly classified copy-move regions, while the FPR f denotes that of incorrectly located cloned regions. They are defined as [10, 18]:

$$t = \frac{N_{\mathrm{TP}}}{N_{\mathrm{TP}}+N_{\mathrm{FN}}}, f = \frac{N_{\mathrm{FP}}}{N_{\mathrm{TN}}+N_{\mathrm{FP}}}, \tag{19}$$

where N_{TP} is the number of correctly detected forged images, N_{TN} indicates the number of correctly detected authentic images, N_{FP} denotes the number of authentic images which have been erroneously detected as forged, and N_{FN} denotes the number of forged images which have not been detected. The accuracy of CMFD schemes a denotes the performance of CMFD schemes based on TPR and FPR. It is defined as below [10, 18]:

$$a = \frac{t+1-f}{2}, \tag{20}$$

However, at the pixel level, we should not only pay attention if the proposed scheme can distinguish forged images and authentic images, but also cover detected forgery regions perfectly. In this case, precision p and recall r [7, 29, 46] are used to evaluate detection performance. Metrics p, r and F_1 are defined as follows [7]:

$$p = \frac{N_{\mathrm{TP}}}{N_{\mathrm{TP}} + N_{\mathrm{FP}}}, r = \frac{N_{\mathrm{TP}}}{N_{\mathrm{TP}} + N_{\mathrm{FN}}}, \tag{21}$$

where N_{TP} denotes the number of correctly detected forged pixels, N_{FP} denotes the number of pixels which has been erroneously detected as forged, and N_{FN} is the number of forged pixels which has not been detected. p is used to describe the percentage of correctly detected pixels. A higher value of p means there are less erroneous detections. r describes whether the forgery areas are completely covered or not. A higher value of r means the more complete the coverage of forgery areas is.

By combining p with r, the F_1 score is obtained [7]. The higher F_1 score gets, the better the performance is.

$$F_1 = 2 \times \frac{p \times r}{p + r}. \tag{22}$$

An intuitive illustration of the relationship between N_{TP}, N_{FP}, and N_{FN} is shown in Fig. 7. As the way of presenting in [28, 44] is clear, we employ the same way: green for correct detected areas, red for incorrect detected areas, and white for ground-truth areas, in which forged areas have not been detected.

4.3 Detection results obtained on FAU at pixel level

In this section, we mainly examined the ability of CMFD schemes to distinguish both authentic and forged images from FAU dataset [7] at pixel level. They should be able to show the forged areas in detail, which means they can perfectly display the particulars in the ideal situation. The performance of the proposed scheme is compared with that of various state-of-the-art CMFD methods, including block-based methods (e.g. [9, 30, 36]), keypoint-based methods (e.g. [1, 14, 23, 33, 42]) and fusion of both (e.g. [29, 44, 46]).

4.3.1 Plain CMFD

We first evaluate their plain copy-move foregery detection performance. The detection results of the 48 images from the *nul* sub-dataset are listed in Table 4, in descending F_1 order. The proposed scheme, while using DCT, achieves the optimal F_1, with $p = 92.61\%$, $r = 91.48\%$, and $p = 91.56\%$. It has better CMFD performance at the pixel level compared with other algorithms. The highest r is achieved when using PCET, because the coverage is more comprehensive; however, this leads to more coverage errors. Wang et al. [19] achieved the highest p, which means they had the least number of detection errors. To sum up, our scheme reaches better results at the image level and the pixel level in the case of plain copy-move forgery.

4.3.2 CMFD under various attacks

As images are not only forged under plain copy-move manipulations, the robustness of different schemes should be tested especially when they are under various attacks.

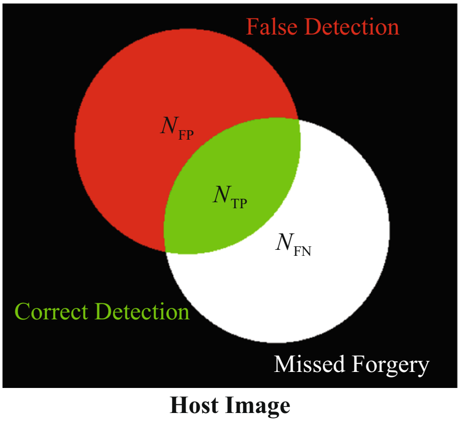


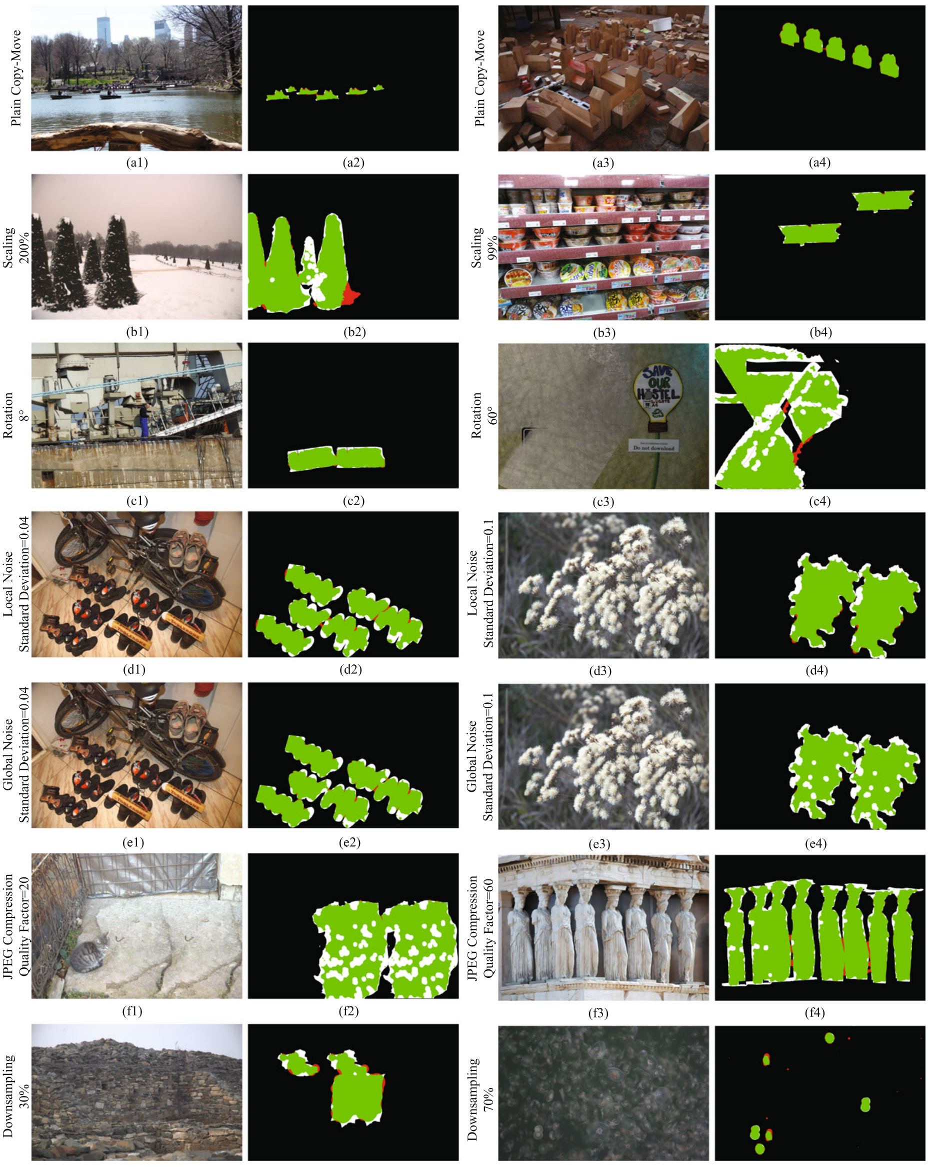
Fig. 7 A visualization of the relationship between N_{TP}, N_{FP}, and N_{FN} [28]

Table 4 Detection results under plain copy-move forgery at the pixel level in descending F_1 order

Schemes	$p(\%)$	$r(\%)$	$F_1(\%)$
ECDC-DCT	92.61	91.48	91.56
Wang [36]	98.69	85.44	90.92
Ryu [30]	95.07	87.72	90.29
Cozzolino [9]	92.98	88.98	90.19
Pun [29]	97.22	83.73	89.97
Zheng [46]	87.32	85.43	86.27
ECDC-PCET	81.84	93.31	86.09
Zandi [44]	83.65	79.53	79.66
SURF [33, 42]	68.13	76.43	69.54
SIFT [1, 14, 23]	60.80	71.48	63.10

Therefore, sub-datasets of various types attacks are used, including scaling, rotation, noise, JPEG compression, downsampling, to present CMFD performance. Figure 8 shows the detection results of our scheme under different attacks. The first and third columns represent forged images. The second and fourth columns are detection results. Figure 8a1 and a3 show the plain copy-move forgery; Fig. 8b1 and b3 show the forged fragments scaled respectively by small and large scaling factors; Fig. 8c1 and c3 show the images under different rotation angles attacks; Fig. 8d1 and d3 show the local noise with two different standard deviations; Fig. 8e1 and e3 are the images under global noise attacks with two different standard deviations; Fig. 8f1 and f3 show the forged images attacked by JPEG compression with two different quality factors; Fig. 8g1 and g3 are two forged images downsampled by different downsampling factors. It can be seen from the results that our scheme also performs well on tampered images or forged fragments under various geometric transformations and signal processing.

Figures 9, 10 and 11 show the p, r, and F_1 at the pixel level under (a) scaling, (b) rotation, (c) local noise, (d) global noise, (e) JPEG compression, and (f) downsampling attacks with different colors for different schemes' results. Figure 9 shows the p results of the proposed scheme compared with the aforementioned schemes under different attacks. We can observe that the precision of the proposed scheme surpasses most of the others. Under small-scale rotation and scaling, the proposed scheme performs well, its precision with DCT is higher than that with PCET. In terms of large-scale rotation and scaling attacks, the results of the proposed scheme are superior to most of the others. The test results are displayed in Fig. 8b2. Remarkably, the precision of our method running with DCT reaches more than 80% at large-scale magnification. It also shows the highest results under the most severe global noise attacks. However, our scheme is affected by JPEG compression because of the extraction of many inoperative keypoints, especially when the quality factor is below 30. In this situation, as shown in Fig. 8f2, our scheme can only maintain good performance to detect large forged areas, as it cannot filter out the invalid matches which are far more than the correct matches, when faced with small tampered areas. Of course, by making parameters of RANSAC and ECDC thresholds more stringent can reduce false coverage and the precision of JPEG compression with a low quality factor can be significantly improved; nevertheless, as the number of effective keypoints decreases, the precision of these results is reduced under local noise and global noise attacks. At this point, after strict filtering and


Plain Copy-Move
Plain Copy-Move
Scaling 200%
Scaling 99%
Rotation 8°
Rotation 60°
Local Noise Standard Deviation=0.04
Local Noise Standard Deviation=0.1
Global Noise Standard Deviation=0.04
Global Noise Standard Deviation=0.1
JPEG Compression Quality Factor=20
JPEG Compression Quality Factor=60
Downsampling 30%
Downsampling 70%

(a1) (a2) (a3) (a4)

(b1) (b2) (b3) (b4)

(c1) (c2) (c3) (c4)

(d1) (d2) (d3) (d4)

(e1) (e2) (e3) (e4)

(f1) (f2) (f3) (f4)

Fig. 8 Visualized CMFD results of the proposed scheme under various attacks where (a1), (a3), (b1), (b3), (c1), (c3), (d1), (d3), (f1), (f3), (g1), and (g3) respectively are center park, bricks, christmas hedge, supermarket, ship number, tapestry, sweets, white, lone cat, kore, stone ghost, and jellyfish chaos. The first row shows the forgery images and the detection results under plain copy-move forgery. The second and third rows show the forgery images and the detection results under scaling and rotation. The fourth and fifth rows show the forgery images and the detection results under local noise and global noise, respectively. The sixth and last rows show the forgery images and the detection results under JPEG compression and downsampling, respectively. Green is the label of correctly detected regions and false areas are indicated in red. White color specifies the ground-truth areas, in which forged areas have not been detected

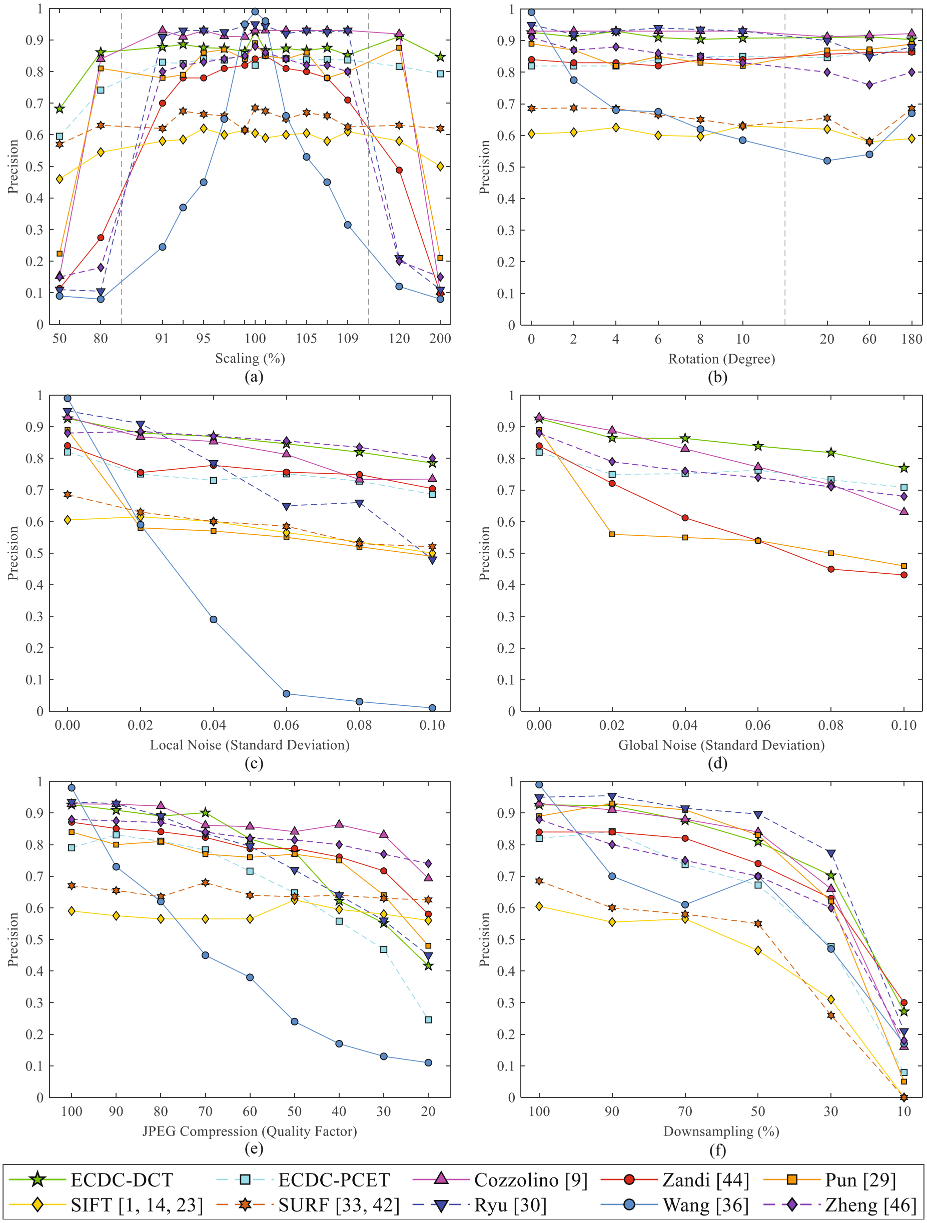


Fig. 9 Precision results at the pixel level: (**a**) Scaling; (**b**) Rotation; (**c**) Local noise; (**d**) Global noise; (**e**) JPEG compression; (**f**) Downsampling

ECDC, there will be only a few remaining matched pairs which do not have the ability to completely cover tampered areas. To sum up, it requires a compromise between performance under serious noise and under JPEG compression with a extremely low quality factor.

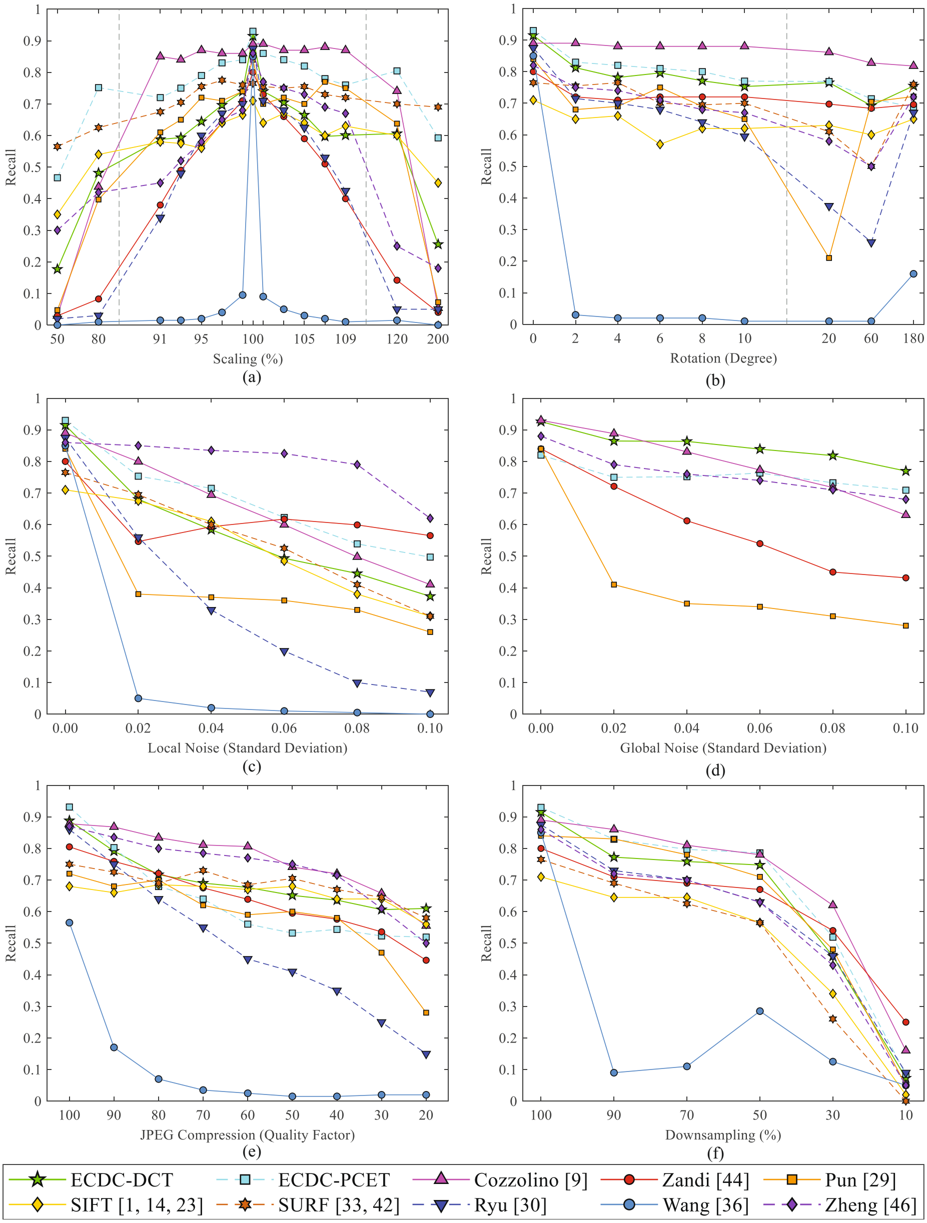


Fig. 10 Recall results at the pixel level: (**a**) Scaling; (**b**) Rotation; (**c**) Local noise; (**d**) Global noise; (**e**) JPEG compression; (**f**) Downsampling

From Fig. 10, it also can be observed that the recall of our scheme with PCET is higher than that with DCT; therefore, we recommend using ECDC with PCET in vulnerable situations to cover forged areas more completely. Considering Figs. 9 and 10, we note that higher recall leads to lower precision, which means that the larger the coverage is, the lower the accuracy of detection may be. If tampered areas only have to be precisely indicated and

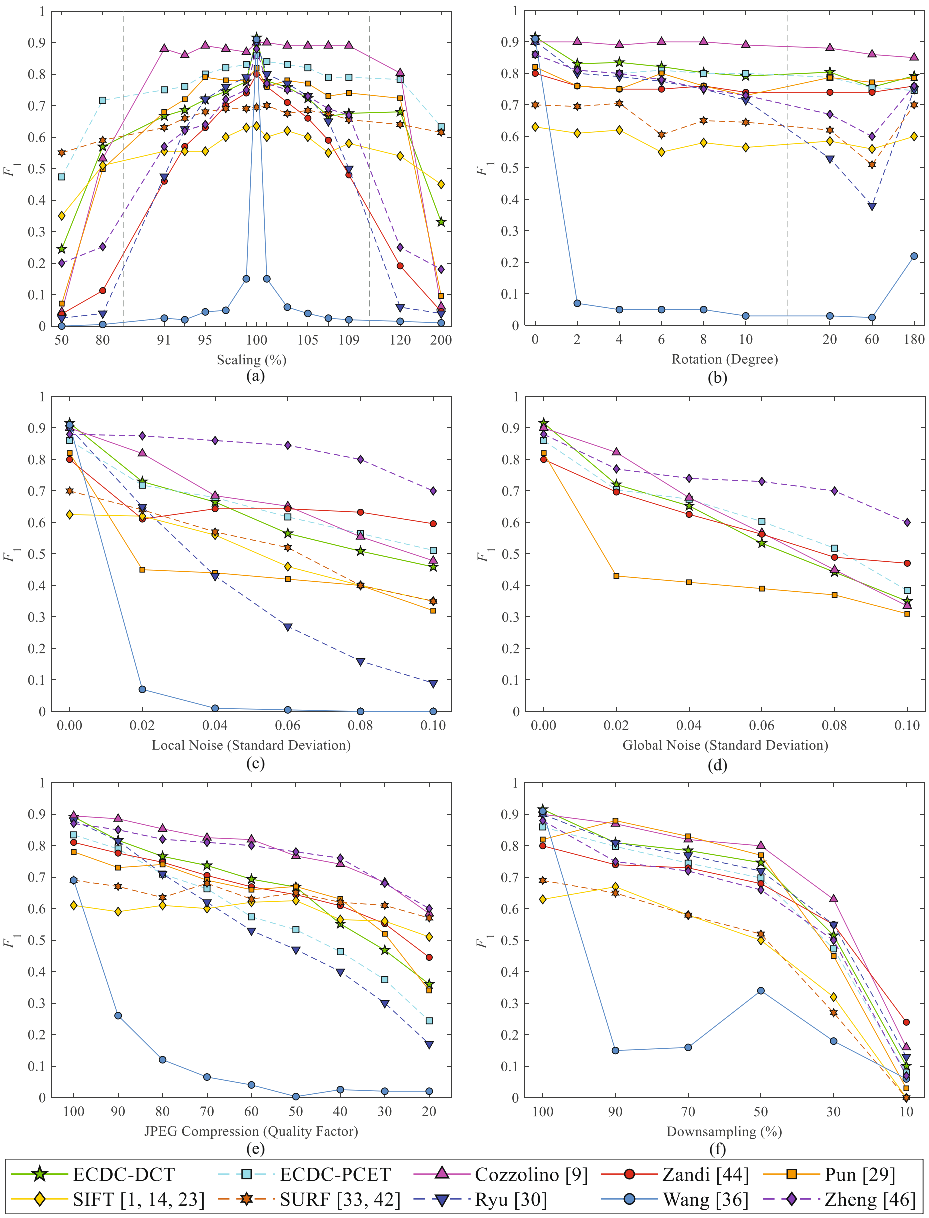


Fig. 11 F_1 scores at the pixel level: (**a**) Scaling; (**b**) Rotation; (**c**) Local noise; (**d**) Global noise; (**e**) JPEG compression; (**f**) Downsampling

they do not need to be presented perfectly, using ECDC with DCT would be a better option because it has higher detection precision with fewer mismatches.

Figure 11 depicts the comparison of all F_1 scores. We can intuitively conclude that ECDC is robust against all kinds of attacks whether it is in combination with DCT or PCET. Though the robustness of ECDC against some attacks is slightly inferior to the scheme [9], it is

Table 5 Running time of the proposed scheme and other schemes in ascending order

Schemes	Running Time (s)
Pun [29]	128.45
Cozzolino [9]	149.03
ECDC-DCT	164.81
Zandi [44]	192.23
ECDC-PCET	275.87
Zheng [46]	554.36

exceptionally better in large-scale detection compared with most classic and state-of-the-art schemes tested.

4.3.3 Running time comparison

To comprehensively evaluate a CMFD scheme, we should pay attention to its running efficiency in addition to its effectiveness and reliability; thus, we also evaluate the processing efficiency of proposed scheme and others on FAU datasets. As the experimental platform of Christlein et al. [7] is different from ours, we only compare the schemes avaiable and implemented on the same platform, and record the average running time of each scheme in Table 5, in an ascending order. It can be observed that our running time is relatively fast, and is above average compared with other solutions.

4.4 Detection results obtained on GRIP and COVERAGE at image level

In this section, for a comprehensive and fair comparison, other popular datasets, such as GRIP [9] and COVERAGE [41], and metrics are used to evaluate state-of-the-art CMFD methods at image level. Figure 12 illustrates serveral challenging forgery detection examples for these two datasets by using proposed method.

GRIP [9] contains some extremely smooth forged regions which is challenging for many keypoint-based methods, such as first three columns of Fig. 12. For comparison, keypoint-based methods [1, 6, 34], block-based methods [5, 9] and fusion of both methods [17, 18, 44] are used. Table 6 presents the detection performance on this dataset, in descending ACC order. As shown in Table 6, both Li [18] and Bravo-Solorio [5] exhibits the highest ACC of

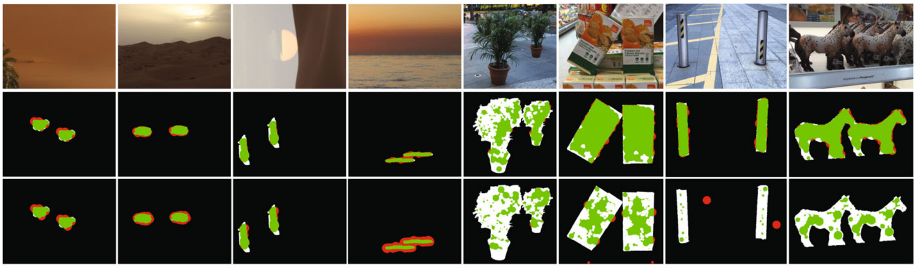


Fig. 12 Some challenging examples of copy-move forgery detection results on GRIP and COVERAGE. From top to bottom: forged images, results obtained by ECDC-DCT, and results obtained by ECDC-PCET. White color indicates the ground-truth; the correct detection pixels are marked in green, while those wrongly are in red

Table 6 The average TPR t, FPR f, and ACC a of detection results on grip at the pixel level in descending a order

Schemes	$t(\%)$	$f(\%)$	$a(\%)$
Li [18]	100	0	100
Bravo-Solorio [5]	100	0	100
ECDC-DCT	97.50	0	98.75
ECDC-PCET	95.00	1.25	96.86
Cozzolino [9]	98.75	8.75	95.00
Chen [6]	90.00	10.42	89.79
Zandi [44]	100	33.75	83.12
Silva [34]	100	38.75	80.63
Amerini [1]	70.00	20.00	75.00
Li [17]	83.75	35.00	74.38

100%. The proposed CMFD algorithm using DCT and PCET achieves the second and third rank repectively, with an ACC of 98.75% and 96.86%. For this dataset, block-based methods [5, 9] and fusion of both methods [17, 18, 44] demonstrate generally better performance than keypoint-based methods [1, 6, 34], due to challenging smooth tampered images.

Each image in COVERAGE [41] contains similar-but-genuine objects, resulting the fact that discrimination of forged from genuine objects is highly challenging. Moreover, many of their images are forged under a combination of image attacks. For comparison, keypoint-based methods [1, 25, 34], block-based methods [5, 9] and fusion of both methods [17, 18, 44] are used. Table 7 shows the detection results on COVERAGE, in descending ACC order. It is obvious that all the algorithms perform poorly on this dataset. Silva [34] achieves the best TPR but the highest FPR high false positive rate, while Bravo-Solorio [5] do not wrongly detected any authentic image as tampered one but it has the lowest TPR. Compared with the other algorithms, our method using DCT obtains the best ACC of 75.50% and using PCET gets the third one.

Table 7 The average TPR t, FPR f, and ACC a of detection results on coverage at the pixel level in descending a order

Schemes	$t(\%)$	$f(\%)$	$a(\%)$
ECDC-DCT	76.00	25.00	75.50
Bravo-Solorio [5]	50.55	0	75.30
ECDC-PCET	69.00	22.00	73.50
Li [18]	80.22	41.76	69.23
Cozzolino [9]	59.34	21.98	68.68
Park [25]	78.00	43.00	67.50
Amerini [1]	85.71	54.95	65.38
Li [17]	87.91	63.74	62.09
Silva [34]	91.21	70.33	59.94
Zandi [44]	76.92	71.43	52.75

5 Conclusion

Nowadays, the phenomenon of easy falsification of images has been a hot spot in the field of digital image forensics and information security. Copy-move forgery is one of the most common manipulations in image forgery. In this paper, we propose a new CMFD scheme based on ECDC. By using the combination of SIFT and LPSD extraction algorithms, we get both SIFT and LPSD descriptors of the entire image. In this way, those descriptors can get more detail features, while being more robust to various attacks. Then we use g2NN to gain a large number of matched pairs. After that, we use RANSAC to eliminate most of the mismatched pairs and gain more precise matched pairs; thus, forgery regions have been located roughly. Then, to get the accurate forgery regions, we propose ECDC algorithm, which can cover forgery regions according to the block features of evolving circular domains. Finally, we use morphological operations to improve the results of ECDC algorithm.

Nowadays, as the resolution of images gets higher, their size gets larger. Due to the complexity of the matching features step, block-based methods take too much time, and keypoint-based methods have difficulty in perfectly covering forgery regions. These two factors become our driving force to propose this scheme. In that way, we surmount the barriers caused by applying block features or keypoints alone.

We conduct a large number of experiments on the proposed scheme with satisfactory results to testify that it is an advanced scheme in CMFD field. Those results show both high effectiveness and efficiency, a notable increase in evaluation metrics and running speed. In comparison with other state-of-the-art CMFD schemes, the proposed scheme achieves more outstanding performance, especially under plain copy-move forgery.

In the future, we will strive to combine ECDC with more robust features, and enable it to cope with more various image attacks. Meanwhile, we are looking for a more flexible and reasonable way of fusing block-based and keypoint-based methods, so that it can get better performance and higher efficiency.

Acknowledgements This work was supported in part by the Shandong Provincial Natural Science Foundation, China (Nos. ZR2021MF060, ZR2017MF020), in part by the Education and Teaching Reform Research Project of Shandong University, Weihai (No. Y2021054), in part by the National Natural Science Foundation of China (No. 61702303), in part by the Science and Technology Development Plan Project of Weihai Municipality in 2020, and in part by the 14th Student Research Training Program (SRTP) at Shandong University, Weihai (No. A19167).

Declarations

Conflict of Interests The authors have no competing interests to declare that are relevant to the content of this article.

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 Inf Forensics Secur 6(3):1099–1110
2. Andrews H, Patterson C (1976) Singular value decompositions and digital image processing. IEEE Trans Acoust Speech Signal Process 24(1):26–53
3. Bay H, Tuytelaars T, Van Gool L (2006) SURF: speeded up robust features. In: Proc. Lect. Notes Comput. Sci. Graz, pp 404–417
4. Bayram S, Sencar HT, Memon N (2009) An efficient and robust method for detecting copy-move forgery. In: Proc. IEEE Int. Conf. Acoust. Speech Signal Process. Taipei, pp 1053–1056
5. Bravo-Solorio S, Nandi AK (2011) Exposing duplicated regions affected by reflection, rotation and scaling. In: Proc. International conference on acoustics, speech and signal processing, pp 1880–1883
6. Chen H, Yang X, Lyu Y (2020) Copy-move forgery detection based on keypoint clustering and similar neighborhood search algorithm. IEEE Access 8:36863–36875
7. 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
8. Cozzolino D, Poggi G, Verdoliva L (2014) Copy-move forgery detection based on PatchMatch. In: Proc. IEEE Int. Conf. Image Process. Paris, pp 5312–5316
9. Cozzolino D, Poggi G, Verdoliva L (2015) Efficient dense-field copy-move forgery detection. IEEE Trans Inf Forensics Secur 10(11):2284–2297
10. Ferreira A, Felipussi SC, Alfaro C, Fonseca P, Vargas-Muñoz JE, Dos Santos JA, Rocha A (2016) Behavior knowledge space-based fusion for copy-move forgery detection. IEEE Trans Image Process 25(10):4729–4742
11. Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24(6):381–395
12. Fridrich AJ, Soukal BD, Lukáš AJ (2003) Detection of copy-move forgery in digital images. In: Proc. Digit. Forensic Res. Workshop. Cleveland, pp 55–61
13. Golub GH, Reinsch C (1970) Singular value decomposition and least squares solutions. Numer Math 14(5):403–420
14. Huang H, Gou W, Zhang Y (2008) Detection of copy-move forgery in digital images using SIFT algorithm. In: Proc. Pacific-Asia Workshop Comput. Intel. Ind. Appl. Wuhan, vol 2, pp 272–276
15. Kaura WCN, Dhavale S (2017) Analysis of SIFT and SURF features for copy-move image forgery detection. In: Proc. Int. Conf. Innov. Inf., Embed. Commun. Syst. Tamil Nadu, pp 1–4
16. Li Y (2013) Image copy-move forgery detection based on polar cosine transform and approximate nearest neighbor searching. Forensic Sci Int 224(1–3):59–67
17. Li J, Li X, Yang B, Sun X (2015) Segmentation-based image copy-move forgery detection scheme. IEEE Trans Inf Forens Secur 10(3):507–518
18. Li Y, Zhou J (2019) Fast and effective image copy-move forgery detection via hierarchical feature point matching. IEEE Trans Inf Forensics Secur 14(5):1307–1322
19. Lowe DG (1999) Object recognition from local scale-invariant features. In: Proc. IEEE Int. Conf. Comput. Vision. Kerkyra, pp 1150–1157
20. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
21. Okawa M (2018) From BoVW to VLAD with KAZE features: offline signature verification considering cognitive processes of forensic experts. Pattern Recogn Lett 113:75–82
22. Okawa M (2018) Synergy of foreground–background images for feature extraction: offline signature verification using Fisher vector with fused KAZE features. Pattern Recogn 79:480–489
23. Pan X, Lyu S (2010) Region duplication detection using image feature matching. IEEE Trans Inf Forensics Secur 5(4):857–867
24. Pandey RC, Singh SK, Shukla KK, Agrawal R (2014) Fast and robust passive copy-move forgery detection using SURF and SIFT image features. In: Proc. Int. Conf. Ind. Inf. Syst., Gwalior. India, Art. no. 7036519
25. Park JY, Kang TA, Moon YH, Eom IK (2020) Copy-move forgery detection using scale invariant feature and reduced local binary pattern histogram. Symmetry 12(4):1–16. article 492
26. Photo Tampering Throughout History. Accessed: Nov. 20, 2019. [Online]. Available: https://pth.izitru.com/2016_02_01.html
27. Popescu AC, Farid H (2004) Exposing digital forgeries by detecting duplicated image regions, Dept. Comput. Sci., Dartmouth College, Hanover, NH, USA, Tech. Rep TR2004-515

28. Pun CM, Chung JL (2018) A two-stage localization for copy-move forgery detection. Inf Sci 463–464:33–55
29. Pun CM, Yuan XC, Bi XL (2015) Image forgery detection using adaptive oversegmentation and feature point matching. IEEE Trans Inf Forensics Secur 10(8):1705–1716
30. Ryu SJ, Lee MJ, Lee HK (2010) Detection of copy-rotate-move forgery using Zernike moments. In: Proc. Lect. Notes Comput. Sci. Alberta, pp 51–56
31. Sharma S, Ghanekar U (2015) A rotationally invariant texture descriptor to detect copy move forgery in medical images. In: Proc. IEEE Int. Conf. Comput. Intell. Commun. Technol. Ghaziabad, pp 795–798
32. Shehab A, Elhoseny M, Muhammad K, Sangaiah AK, Yang P, Huang H, Hou G (2018) Secure and robust fragile watermarking scheme for medical images. IEEE Access 6:10269–10278
33. Shivakumar BL, Baboo SS (2011) Detection of region duplication forgery in digital images using SURF. Int J Comput Sci Issues 8(4–1):199–205
34. 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
35. Tao T, Zhang Y (2017) A scale-invariant keypoint detector in log-polar space. In: Proc. SPIE Int. Soc. Opt. Eng. Tokyo, vol 10225, Art. no. 102250P
36. Wang J, Liu G, Li H, Dai Y, Wang Z (2009) Detection of image region duplication forgery using model with circle block. In: Proc. Int. Conf. Multimedia Inf. Networking Secur, Wuhan, pp 25–29
37. Wang J, Liu G, Zhang Z, Dai Y, Wang Z (2009) Fast and robust forensics for image region-duplication forgery. Acta Auto Sin 35(12):1488–1495
38. Wang X, Xue J, Zheng Z, Liu Z, Li N (2012) Image forensic signature for content authenticity analysis. J Visual Commun Image Represent 23(5):782–797
39. Wang C, Zhang Z, Li Q, Zhou X (2019) An image copy-move forgery detection method based on SURF and PCET. IEEE Access 7:170032–170047
40. Wang C, Zhang H, Zhou X (2018) A self-recovery fragile image water-marking with variable watermark capacity. Appl Sci 8(4):Art. no. 548
41. Wen B, Zhu Y, Subramanian R, Ng TT, Shen X, Winkler S (2016) COVERAGE—a novel database for copy-move forgery detection. In: Proc. IEEE International conference on image processing. Phoenix, pp 161–165
42. Xu B, Wang J, Liu G, Dai Y (2010) Image copy-move forgery detection based on SURF. In: Proc. Int. Conf. Multimedia Inf. Networking Secur. Nanjing, pp 889–892
43. Yap PT, Jiang X, Kot AC (2010) Two-dimensional polar harmonic transforms for invariant image representation. IEEE Trans Pattern Anal Mach Intell 32(7):1259–1270
44. Zandi M, Mahmoudi-Aznaveh A, Talebpour A (2016) Iterative copy-move forgery detection based on a new interest point detector. IEEE Trans Inf Forensics Secur 11(11):2499–2512
45. Zear A, Singh AK, Kumar P (2018) A proposed secure multiple watermarking technique based on DWT, DCT and SVD for application in medicine. Multimed Tools Appl 77(4):4863–4882
46. 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. Multidimens Syst Signal Proc 27(4):989–1005