

Received October 9, 2019, accepted November 4, 2019, date of publication November 22, 2019, date of current version December 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2955308

An Image Copy-Move Forgery Detection Method Based on SURF and PCET

CHENGYOU WANG¹, (Member, IEEE), ZHI ZHANG¹,
QIANWEN LI¹, AND XIAO ZHOU¹, (Member, IEEE)

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

Corresponding author: Xiao Zhou (zhouxiao@sdu.edu.cn)

This work was supported in part by the Shandong Provincial Natural Science Foundation, China, under Grant ZR2017MF020, in part by the National Natural Science Foundation of China under Grant 61702303, in part by the Science and Technology Development Plan Project of Weihai Municipality under Grant 2018DXGJ07, and in part by the Education and Teaching Reform Research Project of Shandong University, Weihai, China, under Grant Y2019032.

ABSTRACT With the maturity of image editing software, image content has been forged frequently, posing potential threats to many critical fields. To detect forgery images effectively, this paper proposes an image copy-move forgery detection (CMFD) method based on speeded-up robust feature (SURF) and polar complex exponential transform (PCET). Firstly, image is divided into non-overlapping irregular image blocks by superpixel segmentation. Then, these image blocks are separated into two categories: smooth regions and texture regions. Secondly, after finding the keypoints by SURF, the PCET coefficients are extracted and utilized for searching similar features by feature matching algorithm. Thirdly, a strategy is used to eliminate false matched points and find the regions with dense matched points. It combines the random sample consensus (RANSAC) algorithm and a filtering scheme. Finally, mathematical morphology and an iterative strategy are adopted to refine the tampered regions. Compared with other CMFD methods, the proposed method can detect the forgery which occurs in high-brightness smooth regions or forgery images involving similar but genuine regions. Experimental results also indicate the proposed method can resist different distortions by various attacks, including rotation, scaling, blurring, joint photographic expert group (JPEG) compression, and noise addition.

INDEX TERMS Image forensics, image copy-move forgery detection (CMFD), speeded-up robust feature (SURF), polar complex exponential transform (PCET), superpixel segmentation.

I. INTRODUCTION

With the rapid development of the Internet, it becomes easy to obtain abundant multimedia information [1]. It is convenient for people to get high-resolution pictures and videos with their cameras or mobile phones, enriching their lives. However, people can alter the content of images as their wishes using various image editing software arbitrarily, such as Adobe PhotoShop [2] and ACDSee Photo Editor [3]. The authenticity and integrity of images have been threatened in many critical fields [4]–[7]. For example, forged medical films may cause misdiagnosis and affect the state of illness [5], and forged newspaper photographs may mislead people and cause social turbulence [6]. Therefore, image forensics technique as a significant part of information

security, which aims at identifying the forgery, is urgent to be developed [4]–[7].

In recent decades, scholars have proposed different methods to distinguish between original images and forgery images, which are divided into active forensics and passive forensics [1]. Active forensics techniques are used to verify the integrity of the verification information such as digital watermark [8]–[10] and digital signature [11]–[13]. Active forensics techniques have the advantages of strong detection ability and are not easy to be avoided. However, in active forensics techniques, the verification information needs to be inserted into carrier images before distribution, which decreases the quality of images. Passive forensics techniques are used to verify the authenticity by analyzing the information and structure of images, which overcome the defects of active forensics techniques.

There are mainly two forgeries to alter the content of images: splicing and copy-move [1]. Splicing forgery is a

The associate editor coordinating the review of this manuscript and approving it for publication was Haiyong Zheng¹.

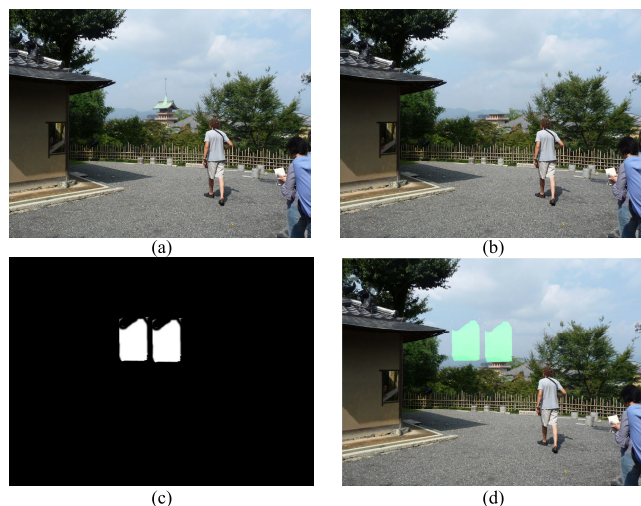


FIGURE 1. Example of image copy-move forgery on image Japan tower: (a) original image, (b) forgery image, (c) tampered region, and (d) forgery image marked with the tampered region.

way to copy and paste a part of an image into another image. Copy-move forgery is a way to copy and paste a part of an image into the same image. An example of image copy-move forgery on image Japan tower is given in Fig. 1. Fig. 1 shows an original image, corresponding tampered region, and forgery image marked with the tampered region by green. In Fig. 1(b), the Japan tower is concealed using a part of the sky region within the same image. The forgery of the Japan tower in Fig. 1(b) is difficult to be recognized because the pasted sky region has similar characteristics with the whole image. Therefore, image copy-move forgery detection (CMFD) is a challenging topic.

Image CMFD methods are mainly divided into two categories: block-based methods and keypoint-based methods [14].

In block-based CMFD methods [15]–[26], an image is divided into many image blocks from which features are extracted. The existence of similar features is the basis of judging whether the image is tampered with. At present, the block-based image forgery detection methods can locate the tampered regions of forgery images accurately. However, these methods have high computational complexity, and are difficult to resist large-scale rotation and scaling.

In keypoint-based CMFD methods [27]–[40], keypoints are extracted from an image, and then the descriptors of these keypoints are used in feature matching. To determine whether the image is tampered with, the number of matched keypoints should be compared with the pre-set threshold. Most of current keypoint-based CMFD methods have difficulty in judging whether smooth regions of image are tampered with correctly. Compared with block-based CMFD methods, keypoint-based CMFD methods have lower computational complexity.

This paper proposes an image CMFD method to solve the problem that most of keypoint-based CMFD methods are

difficult to detect forgery which occurs in smooth regions or in forgery images involving similar but genuine regions. Firstly, to narrow down the search range of feature matching, the image block classification technique is investigated. Hence, the image blocks are divided into smooth regions and texture regions, and the tampered regions belong to the same type of image regions. Secondly, a filtering scheme, based on the number of matched points in the statistical image block, is used to eliminate false matched points. The proposed CMFD method combines the advantages of block-based and keypoint-based image CMFD methods. Finally, mathematical morphology and an iterative strategy are adopted to refine tampered regions.

The rest of this paper is organized as follows. Section II reviews related work of CMFD. Section III introduces speed-up robust feature (SURF) and polar complex exponential transform (PCET) used in the proposed method. Section IV presents the proposed image CMFD method in detail. A series of experiments are conducted and discussed to demonstrate the effectiveness of the proposed method in Section V. Section VI gives conclusions and remarks on possible future work.

II. RELATED WORK

In the field of block-based CMFD, Fridrich *et al.* [15] first proposed the CMFD algorithm, which used the discrete cosine transform (DCT) coefficients as the feature to verify the authenticity of an image. It is a landmark work in the field. However, this method has extremely high computational complexity. To reduce its complexity, Huang *et al.* [16] truncated the quantified DCT vector using a constant to reduce the feature dimensionality. In addition, the lexicographical order was used to make the feature matrix orderly, which narrowed the scope of feature searching. To resist various image processing operations, singular value decomposition (SVD) [17], [18], is applied to CMFD methods to get stable features. After extensive research, Luo *et al.* [19] found that a single natural image was unlikely to have two extremely similar regions, whose sizes were larger than 0.85% of the image size. The image is considered as a forgery image if the frequency of shift-vector outperforms pre-set threshold in the methods [15]–[19].

However, there is still a problem that the methods [15]–[19] cannot locate the tampered regions if the tampered regions are rotated or scaled. To solve the problem in [15]–[19], many local invariant features are applied to the image CMFD methods, such as scale-invariant features: Hu moment [20], and rotation-invariant features: Zernike moment (ZM) [21], local binary pattern (LBP) [22], and discrete analytical Fourier-Mellin transform (DAFMT) [23]. polar harmonic transform (PHT) [24], which is a rotation-invariant feature and has low computational complexity, is also used to solve the problem [25], [26].

Since most of block-based CMFD methods have high computational complexity, some keypoint-based CMFD methods are emerging. Amerini *et al.* [27] used the scale invariant

feature transform (SIFT) and its descriptors as features to find similar features by the generalized 2 neighbor nearest (g2NN) algorithm. And then, random sample consensus (RANSAC) algorithm was used to estimate affine transformation matrix and remove false matched points. Two years later, on the basis of the work in [27], Amerini *et al.* [28] used the J-linkage clustering algorithm to detect possible tampered regions. The disadvantage of this method is that the SIFT descriptor has a high dimensionality. Therefore, other keypoint detectors and descriptors are applied to the image CMFD methods, such as scale- and rotation- invariant features: SURF [29], and rotation-invariant features: Harris corner [30], [31], accelerated-KAZE (A-KAZE) [32], oriented FAST and rotated BRIEF (ORB) [33], and multi-support region order-based gradient histogram (MROGH) [34]. Binary robust invariant scalable keypoints (BRISK) [35] is also used due to its scaling and rotation invariance.

However, the above methods [27]–[33] just mark the tampered regions by matched keypoints and their connections. Such identification methods can only identify the approximate regions, and there is a problem that the positioning is inaccurate. To solve the problem, other methods were used in [36]–[40]. Pan and Lyu [36] calculated the correlation coefficients to obtain a correlation coefficient map, using the affine transformation matrix estimated by the RANSAC algorithm. The map and morphological operations were used to locate the tampered regions. Yang *et al.* [37], using the estimated affine transformation matrix, calculated the zero mean normalized cross-correlation (ZNCC) coefficients to locate the tampered regions. However, it is difficult to detect enough keypoints in smooth regions. To obtain enough keypoints, Jin and Wan [38] set the contrast threshold of the SIFT detector to 0. Zandi [39] improved the keypoint extraction method and used multiple iterations to locate the tampered region in smooth regions. The methods [38] and [39] can make accurate detection in smooth regions, but cannot detect the forgery which occurs in forgery images involving similar but genuine regions.

Compared with the block-based CMFD methods, the keypoint-based CMFD methods reduce computational complexity. However, most of the keypoint-based CMFD methods cannot detect the forgery which occurs in high-brightness smooth regions or forgery images involving similar but genuine regions correctly, which is resolved mainly in this paper.

In addition, Pun *et al.* [40] combined the block-based methods and keypoint-based methods. They used the simple linear iterative clustering (SLIC) superpixel segmentation algorithm to divide images into non-overlapping irregular image blocks. The features of each image block were the extracted SIFT features from keypoints in the block, and were used in features matching. However, the positioning effect of this method primarily depends on the selection of the initial value of the SLIC superpixel segmentation algorithm.

With the development of high-performance computing, deep learning has begun to be used in the field of image forensics [41]. However, such method has many shortcomings,

such as a large number of training samples and long training time. At present, the forensic method based on deep learning has a poor positioning effect on the tampered regions of single copy-move forgery image, and the application of deep learning in the field still needs to be explored.

III. TWO FEATURES: SURF AND PCET

This section introduces two features, SURF and PCET, used in the proposed method.

A. SURF

SURF [42] is usually adopted in computer vision, such as image registration and object recognition. SURF, developing on classical SIFT, can not only maintain the scaling and rotation invariance of SIFT, but also be robust to noise, detection displacements, and geometric and illuminated deformations.

SURF determines keypoints by calculating the relevant Hessian matrix and finding the extreme points of scale space. Given a point $\mathbf{x} = (x, y)$ in an image \mathbf{I} , the Hessian matrix $\mathbf{H}(\mathbf{x}, \sigma)$ in \mathbf{x} at scale σ is represented as follows [42]:

$$\mathbf{H}(\mathbf{x}, \sigma) = \begin{bmatrix} C_{xx}(\mathbf{x}, \sigma) & C_{xy}(\mathbf{x}, \sigma) \\ C_{xy}(\mathbf{x}, \sigma) & C_{yy}(\mathbf{x}, \sigma) \end{bmatrix}, \quad (1)$$

where $C_{xx}(\mathbf{x}, \sigma)$ is the convolution of the Gaussian second-order partial derivative $\partial^2 G(x, y, \sigma)/\partial x^2$ with the image at pixel \mathbf{x} . $C_{xy}(\mathbf{x}, \sigma)$ and $C_{yy}(\mathbf{x}, \sigma)$ are similar to the $C_{xx}(\mathbf{x}, \sigma)$, which means that $C_{xy}(\mathbf{x}, \sigma)$ is the convolution of $\partial^2 G(x, y, \sigma)/\partial x \partial y$ with the image and $C_{yy}(\mathbf{x}, \sigma)$ is the convolution of $\partial^2 G(x, y, \sigma)/\partial y^2$ with the image. Fig. 2(a) shows the Gaussian second-order partial derivative in horizontal, vertical, and diagonal directions, respectively.

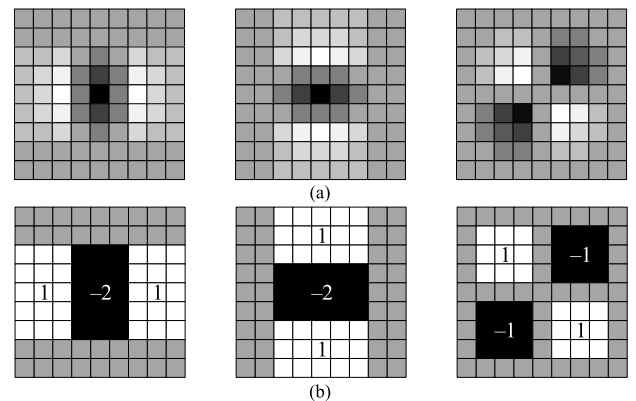


FIGURE 2. The box filters used in SURF, taking 9×9 box filters as an example: (a) the Gaussian second-order partial derivative in x -, y -, and xy - directions, respectively, and (b) the approximation of (a) in x -, y -, and xy - directions, respectively. The grey regions are equal to 0.

Due to the high time complexity of calculating the second-order derivative of the image, SURF uses rectangular box filters to approximate the Gaussian second-order derivative. Fig. 2(b) shows the box filters in horizontal, vertical, and diagonal directions, respectively. The utilization of box filters accelerates the speed of convolution calculation and

reduces complexity. Therefore, the determinant of Hessian matrix \mathbf{H} is approximated as follows [42]:

$$\det(\mathbf{H}) \approx B_{xx}B_{yy} - (0.9B_{xy})^2, \quad (2)$$

where the B_{xx} , B_{yy} , and B_{xy} are the convolution of box filters with the image at pixel \mathbf{x} in horizontal, vertical, and diagonal directions, respectively.

A point is compared with 26 points around the point in a $3 \times 3 \times 3$ neighborhood which is between the 3×3 rectangle windows of the neighbor scales and the current scale. A detected point is considered to be a keypoint, if the point is the extreme point and its determinant of Hessian matrix is higher than the pre-set threshold T_H . Inspired by [38], the experiments are conducted by setting the contrast threshold to 0 with different detectors, such as Harris corner, SURF, SIFT, ORB, and BRISK in OpenCV library. Fig. 3 shows the detection results on image Japan tower, and the digit in subtitles (b), (c), (d), (e), and (f), is the number of detected keypoints by the corresponding detector. In Fig. 3, the red points are the extracted keypoints from the image using Harris corner, SURF, SIFT, ORB, and BRISK, respectively. On the one hand, if the number of detected points is large, the computation time will be long. On the other hand, if the number of detected points is small, some forgeries will be omitted. Fig. 3 shows that Harris corner, SURF, and SIFT detectors can obtain sufficient points uniformly covering the whole image. SURF is appropriate to be chosen as the keypoints detector, due to the moderate number of detected points.

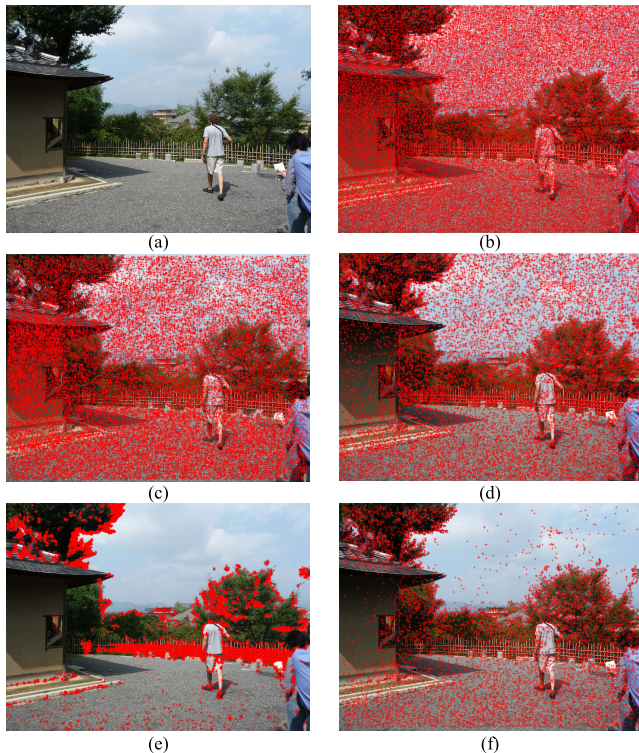


FIGURE 3. Detection results with different detectors on image Japan tower: (a) original image, (b) Harris corner (35040), (c) SURF (31128), (d) SIFT (24356), (e) ORB (20000), and (f) BRISK (9802).

B. PCET

PCET is one form of PHT. PHT including PCET, polar cosine transform (PCT), and polar sine transform (PST), was proposed by Yap *et al.* [24]. PHT has the advantages of Zernike moment's orthogonality and invariance. Moreover, the computation of its kernel function F is much simpler than that of Zernike moment. Therefore, it can be used in where maximal discriminant information is needed.

Since PHT is defined on unit circle, an image $I(x, y)$ in Cartesian coordinates needs to be converted into polar coordinates to obtain $I^P(r, \theta)$, where r and θ are expressed as follows:

$$r = \sqrt{x^2 + y^2}, \quad \theta = \arctan(y/x). \quad (3)$$

The PHT coefficient M_{nl} of n order with l repetition, where $|n| = |l| = 0, 1, \dots, \infty$, is defined in continuous form as follows [24]:

$$M_{nl} = \Omega_n \int_0^{2\pi} \int_0^1 [F_{nl}^P(r, \theta)]^* I^P(r, \theta) r dr d\theta, \quad (4)$$

where $[\cdot]^*$ is the complex conjugate operation. For PCET, $\Omega_n = 1/\pi$. For PST or PCT, Ω_n is defined as (5). $F_{nl}^P(r, \theta)$ is the PHT kernel which is given in (6). Ω_n and $F_{nl}^P(r, \theta)$ are expressed as follows [24]:

$$\Omega_n = \begin{cases} 1/\pi, & n = 0, \\ 2/\pi, & n \neq 0, \end{cases} \quad (5)$$

$$F_{nl}^P(r, \theta) = \begin{cases} e^{i2\pi nr^2} e^{il\theta}, & \text{PCET,} \\ \cos(\pi nr^2) e^{il\theta}, & \text{PCT,} \\ \sin(\pi nr^2) e^{il\theta}, & \text{PST.} \end{cases} \quad (6)$$

In the reconstruction of PHT, to limit the number of its coefficients, the constraint condition $|n| + |l| < L$ is usually adopted, where L is a pre-set value [24]. For discrete implementation, (4) can be rewritten in Cartesian coordinates as follows [24]:

$$M_{nl} = \Omega_n \iint_{x^2+y^2 \leq 1} [F_{nl}(x, y)]^* I(x, y) dx dy, \quad (7)$$

where $F_{nl}(x, y) = F_{nl}(r \cos \theta, r \sin \theta) \equiv F_{nl}^P(r, \theta)$ and $I(x, y) = I(r \cos \theta, r \sin \theta) \equiv I^P(r, \theta)$. An image with the size of $W \times H$, is defined on a discrete domain $g[a, b]$, where $a = 0, 1, \dots, W-1$ and $b = 0, 1, \dots, H-1$. The image is mapped to a domain of $(x_a, y_b) \in [-1, 1] \times [-1, 1]$ with [24]:

$$x_a = \frac{a - W/2}{W/2}, \quad y_b = \frac{b - H/2}{H/2}, \quad (8)$$

where x_a and y_b indicate the mapped domain, subject to $x_a^2 + y_b^2 \leq 1$. The PHT coefficient M_{nl} on discrete domain

TABLE 1. Moments of PCET coefficients of original image Lena and its distorted versions.







| Moment |  |  |  |  |  |  |
|----------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| | Original | Rotation (35°) | Scaling (0.9) | Blurring (2) | JPEG (80) | Noise (0,0.0001) |
| M_{00} | 124.8675 | 124.8598 | 124.8920 | 124.8676 | 124.8592 | 124.8695 |
| M_{01} | 11.3255 | 11.3183 | 11.3304 | 11.3224 | 11.3245 | 11.3292 |
| M_{02} | 5.2121 | 5.2099 | 5.2348 | 5.2071 | 5.2114 | 5.2112 |
| M_{03} | 6.3872 | 6.3821 | 6.3845 | 6.3864 | 6.3847 | 6.3879 |
| M_{10} | 2.6324 | 2.6305 | 2.6573 | 2.6305 | 2.6296 | 2.6282 |
| M_{11} | 6.9405 | 6.9413 | 6.9346 | 6.9428 | 6.9378 | 6.9426 |
| M_{12} | 6.9984 | 6.9966 | 7.0034 | 7.0020 | 6.9955 | 6.9971 |
| M_{20} | 1.0217 | 1.0254 | 1.0998 | 1.0205 | 1.0155 | 1.0204 |
| M_{21} | 4.3672 | 4.3655 | 4.3657 | 4.3700 | 4.3622 | 4.3690 |
| M_{30} | 0.8060 | 0.8052 | 0.8685 | 0.8095 | 0.7968 | 0.8080 |

TABLE 2. The difference between the features of the original image Lena and those of the distorted versions.

| Feature | Rotation (35°) | Scaling (0.9) | Blurring (2) | JPEG (80) | Noise (0,0.0001) | Computation time (s) |
|-----------|----------------|---------------|---------------|---------------|------------------|----------------------|
| Hu [20] | 0.0581 | 0.0971 | 0.0229 | 0.1600 | 0.0519 | 0.4406 |
| ZM [21] | 0.0606 | 0.0421 | 0.0097 | 0.0230 | 0.0054 | 0.0165 |
| LBP [22] | 0.0004 | 0.1335 | 0.1083 | 0.6482 | 0.3116 | 1.3748 |
| PCET [24] | 0.0033 | 0.0233 | 0.0024 | 0.0041 | 0.0020 | 0.0146 |

can be described as follows [24]:

$$\begin{aligned}
 M_{nl} &= \Omega_n \sum_{a=0}^{W-1} \sum_{b=0}^{H-1} [F_{nl}(x_a, y_b)]^* I(x_a, y_b) \Delta x \Delta y \\
 &= \frac{4\Omega_n}{WH} \sum_{a=0}^{W-1} \sum_{b=0}^{H-1} [F_{nl}(x_a, y_b)]^* I(x_a, y_b), \quad (9)
 \end{aligned}$$

where $I(x_a, y_b) = g[a, b]$, $\Delta x = 2/W$, and $\Delta y = 2/H$. More detailed information can also be found in [24].

To demonstrate the invariance of PCET coefficients in geometric transformation and image processing techniques, the PCET coefficients are extracted from the original image and distorted images. Table 1 lists the PCET coefficients of image Lena with the size of 512×512 and those of distorted versions by various attacks. The attacks include rotation with an angle of 35, scaling with a factor of 0.9, blurring with a filter size of 2, joint photographic expert group (JPEG) compression with a quality factor (QF) of 80, and Gaussian noise addition with 0 mean and 0.0001 variance. In Table 1, the inner circle region of image Lena is used to represent the whole image for reducing the effects of other

distortion operations. Table 1 shows the moments / coefficients of PCET have little change under different distortion operations, which demonstrates that the PCET coefficients are suitable for the feature to find similar features in the method of image CMFD.

To evaluate the performance of PCET feature, PCET is compared with other features, including Hu moment [20], ZM [21], and LBP [22]. Table 2 shows the differences between the features of the original image Lena and those of the distorted versions. For example, the difference D_{PCET}^r between the PCET coefficients of the original image and those of the image rotated by 35°, which is 0.0033 in Table 2, is obtained by:

$$D_{PCET}^r = \frac{1}{N} \sum_n \sum_l |M_{nl} - M_{nl}^r|, \quad (10)$$

where M_{nl} , obtained by (9), is the PCET coefficient of n order with l repetition for the original image and shown in Table 1, and M_{nl}^r represents the PCET coefficient for the image rotated by 35°. N is the total number of PCET coefficients M_{nl} . For PCET, there are $N = 10$ features,

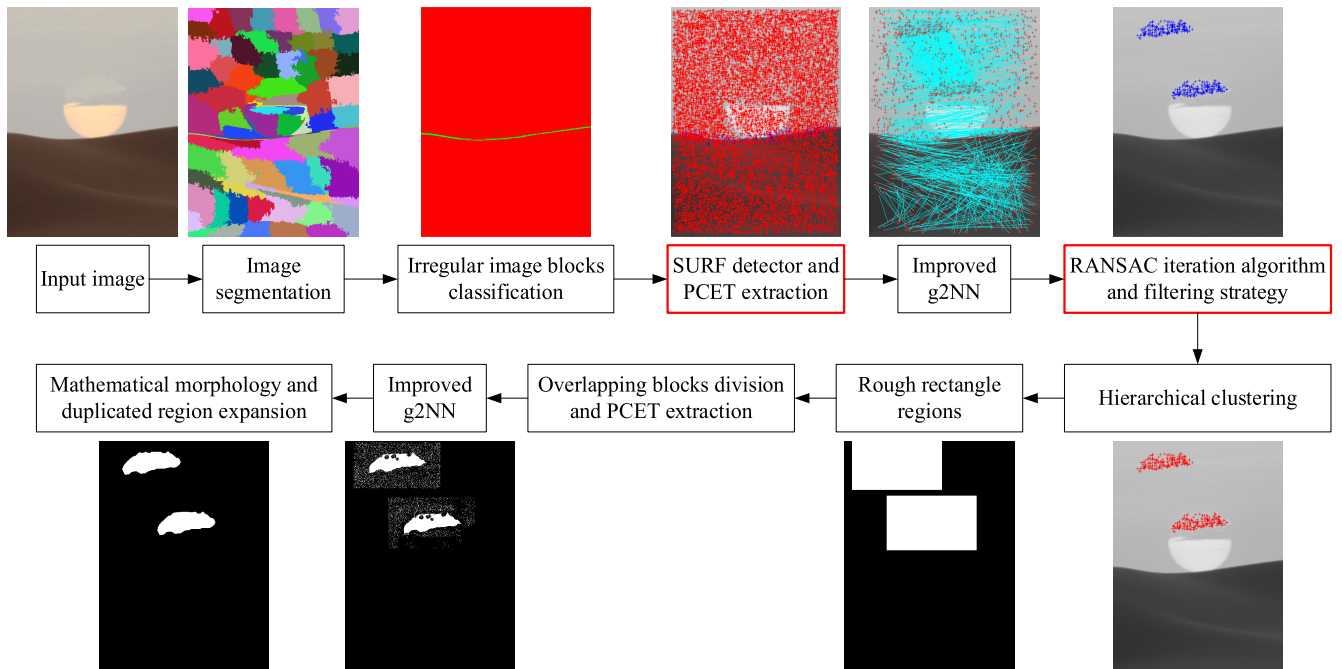


FIGURE 4. Flow chart of the proposed method.

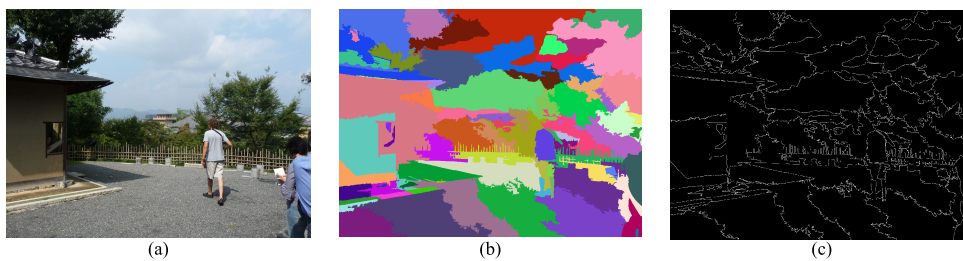


FIGURE 5. Segmentation result on image Japan tower: (a) original image, (b) non-overlapping image blocks with different colors, and (c) edge of the blocks in (b).

including $M_{00}, M_{01}, M_{02}, M_{03}, M_{10}, M_{11}, M_{12}, M_{20}, M_{21}$, and M_{30} . In addition, the information of Hu moment, ZM, and LBP can be found in [20]–[22], respectively. Their difference in Table 2 can be obtained by (10), just like PCET. In Table 2, the computation time of feature extraction in the original image Lena is also listed, which shows that PCET is the fastest. Table 2 demonstrates that PCET is the best one to find similar features in the method of image CMFD.

IV. PROPOSED CMFD METHOD

This section presents the proposed image CMFD method in detail. Its flow chart is given in Fig. 4. Firstly, an image is divided into non-overlapping irregular image blocks, and then the blocks are divided into smooth regions and texture regions. Secondly, the SURF detectors with different contrast thresholds are performed on smooth regions and texture regions to obtain sufficient points. The PCET coefficients of the points are extracted and used as descriptors. An improved g2NN algorithm is proposed and used to search similar features, and the matched points are obtained. Thirdly, the

RANSAC iteration algorithm and a filtering strategy that combines the label matrix are used to eliminate false matched points. The rough rectangle regions are found by dense points. Then, these rectangle regions are divided into overlapping circle blocks and the PCET coefficients are extracted from the circle blocks. The similar features are found by the improved g2NN algorithm. Finally, mathematical morphology and an iterative strategy are used to locate the tampered regions.

A. PRE-PROCESSING

In this method, inspired by block-based CMFD methods, the image is divided into non-overlapping irregular image blocks using minimum barrier superpixel (MBS) segmentation [43]. MBS segmentation is a superpixel segmentation algorithm based on the minimum barrier. Compared with other segmentation methods, MBS segmentation can be configured to make a simple trade-off between performance and efficiency, and is easy to be controlled by one parameter [43]. An example of MBS segmentation result on image Japan tower is given in Fig. 5: Fig. 5(a) is original image, Fig. 5(b) is

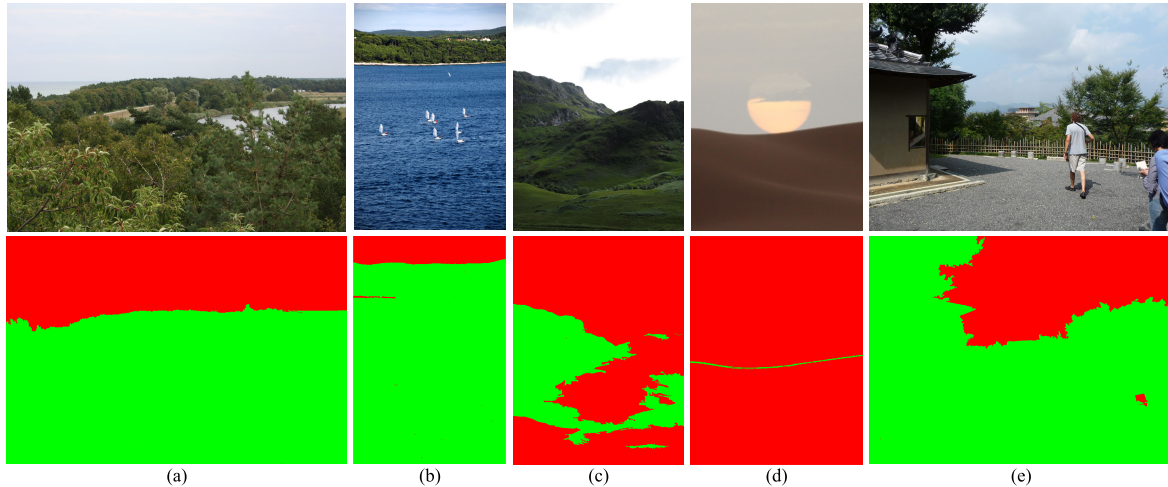


FIGURE 6. Classification results on five images. First row: original images; Second row: the classification results. (a) Beach wood, (b) Sailing, (c) Scotland, (d) TP_C01_023, and (e) Japan tower.

non-overlapping irregular image blocks with different colors, and Fig. 5(c) is edge of the blocks in Fig. 5(b). In Fig. 5(b), different colors indicate different labels in label matrix M_L obtained by MBS segmentation. The label matrix M_L is used to eliminate false matched points in the subsequent steps. There is an inevitable problem to set the initial size of the MBS segmentation algorithm. Based on the discovery of Luo *et al.* [19] and extensive experiments, setting the initial size of image blocks to $W \times H \times 0.01$ is suitable for an image with the size of $W \times H$ in the pre-processing stage.

To narrow the scope of searching similar features, the irregular image blocks are divided into two categories: smooth regions and texture regions. Searching similar features in smooth regions and texture regions respectively will save more time than that in the scope of the whole image. The smooth regions and texture regions are separated according to local information entropy of the irregular blocks.

Supposing image I of size $W \times H$ is divided into K irregular non-overlapping image blocks $R_k (k = 1, 2, \dots, K)$. To avoid the effect of different distortions by various attacks, such as noise addition and blurring, the pixel values are arranged in several pixel intervals determined by the maximum pixel value V_{max} and minimum pixel value V_{min} with 2 pixels. The local information entropy E_k of irregular block R_k is obtained by [44]:

$$E_k = - \sum_i [P_{R_k}(i) \log_2 P_{R_k}(i)], \quad (11)$$

where $P_{R_k}(i)$ is the probability of the i -th pixel interval mentioned above in irregular block R_k which is defined as follows:

$$R_k = \begin{cases} 1, & E_k < E_T, \\ 0, & E_k \geq E_T, \end{cases} \quad (12)$$

where $R_k = 1$ denotes that the irregular block belongs to smooth regions, and $R_k = 0$ denotes that it belongs to texture regions. E_T is a threshold to distinguish the smooth regions

and texture regions. Through experiments, E_T is defined as follows:

$$E_T = E_{min} + \frac{2}{3}E, \quad E = E_{max} - E_{min}, \quad (13)$$

where E_{max} and E_{min} are the maximum and minimum local information entropy among all irregular blocks, respectively. E is the difference value between E_{max} and E_{min} . Therefore, E_T can be written as follows:

$$E_T = \frac{1}{3}E_{min} + \frac{2}{3}E_{max}. \quad (14)$$

According to (12) and (14), the whole irregular blocks are separated into smooth regions and texture regions. For example, five images and their corresponding classification results are shown in Fig. 6. The regions of the sky, desert, and cloud in these images are deemed as smooth regions. The red regions are smooth regions and the green regions are texture regions in the classification images depicted in the second row in Fig. 6. As we can see, the classification results in Fig. 6 regard the regions of the sky, desert, and cloud as smooth regions and regard other regions as texture regions. The examples demonstrate the effectiveness of this classification strategy to distinguish smooth regions and texture regions. The subtitles of Fig. 6 are the image names in the datasets [26], [45].

B. KEYPOINT DETECTION AND DESCRIPTION

The SURF detector is applied to extract keypoints on smooth regions and texture regions. The contrast thresholds T_{SURF}^s in smooth regions and T_{SURF}^t in texture regions are set to different values to obtain sufficient points uniformly covering the whole image. The detection results obtained by the SURF detector on three images are shown in Fig. 7. In Fig. 7, the red points are extracted from smooth regions and the blue points are extracted from texture regions. Then, the square block centered at each point with R_{PCET} pixels is obtained,

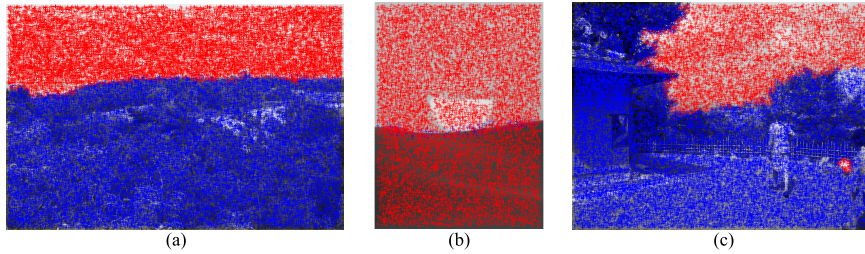


FIGURE 7. Detection results obtained by the SURF detector on three images: (a) Beach wood, (b) TP_C01_023, and (c) Japan tower.

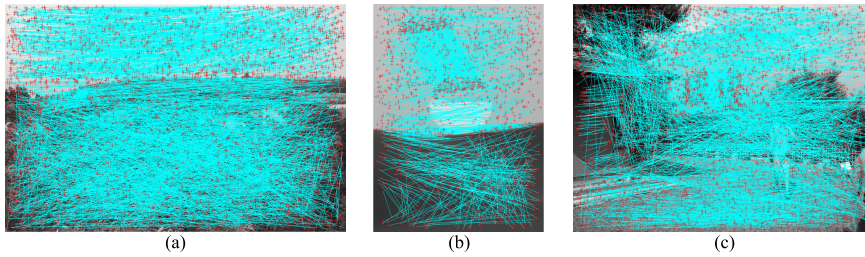


FIGURE 8. Detection results of the improved g2NN algorithm on three images: (a) Beach wood, (b) TP_C01_023, and (c) Japan tower.

which also means that the length of the square block is $(2 \times R_{PCET} + 1)$. As mentioned in Section III.B, the inner circle region of the square block is used as the circle block of the point. The PCET coefficients are extracted from the circle blocks and used as the descriptor of the keypoints.

C. FEATURE MATCHING AND FALSE MATCHED POINTS ELIMINATION

After feature extraction, two group features, f^s and f^t , are obtained. In feature matching phase, the improved g2NN algorithm is performed to each group to search similar features. Taking the smooth group feature f^s for an example, the ordered distance vector $\mathbf{D} = \{d_1, d_2, \dots, d_{n-1}\}$ is obtained by calculating f_i^s and other descriptors using Euclidean distance and making them orderly. If $d_j/d_{j+1} < T_{g2NN}$ in \mathbf{D} is not satisfied, where T_{g2NN} is the pre-set ratio threshold in the g2NN algorithm, the j points are deemed as the candidate points of the x_i^s point [27]. However, since the contrast threshold of the SURF detector is extremely small, the extracted keypoints, whose spatial distance is too short, may be misjudged as matched point pairs. Therefore, it is necessary to improve the g2NN by eliminating the points, whose spatial distance to the point x_i^s is shorter than the pre-set distance threshold T_d .

After performing the improved g2NN algorithm on each point in the smooth group, the matched points in the smooth group x_m^s are obtained. Similarly, the matched points in the texture group x_m^t are also obtained. The improved g2NN algorithm can make the proposed method effective in multiple copy-move forgeries. In this paper, the k-dimensional tree is built based on each feature group, and k-nearest neighbor (KNN) is used to find the N_{KNN} nearest points

of the point. The improved g2NN algorithm is applied to the scope of the N_{KNN} points. After the improved g2NN algorithm, the matched points x_m are obtained by combining matched points in the smooth group x_m^s and texture group x_m^t . The detection results of the improved g2NN algorithm on three images are shown in Fig. 8. In Fig. 8, the red points are the matched points, and they are connected by blue lines.

After the feature matching phase, many false matched points are existing in the matched points x_m . The RANSAC iteration algorithm based on [36] and a filtering strategy combining the label matrix obtained by MBS segmentation are used to find the regions with dense points and eliminate false matched points. RANSAC divides the matched points x_m into inlier and outlier groups. To obtain sufficient matched points, the iteration of RANSAC is stopped only when the number of the points belonging to the inlier group is less than the pre-set threshold T_{in} . The smaller the value of T_{in} , the more the matched points and the longer the running time. The detail of the RANSAC iteration is described in **Algorithm 1**. Next, the frequency of different labels corresponding to the matched points x_m are counted. Then, the points whose corresponding label frequency is less than the pre-set threshold T_L are eliminated from the matched points x_m . Finally, the regions with dense points are found. The detection results which are demarcated by dense points on three images are given in Fig. 9. In Fig. 9, the red points are the final matched points.

D. TAMPERED REGION LOCALIZATION

For each region with dense points, the minimum and maximum coordinates in x and y directions $x_{min}^i, y_{min}^i, x_{max}^i$, and y_{max}^i are found. The rectangle region is determined by

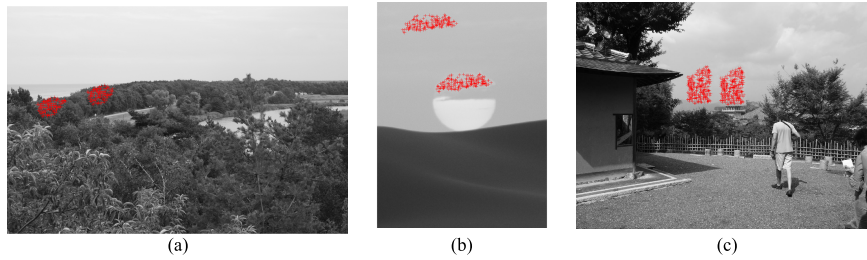


FIGURE 9. Detection results of the rough region located at dense points on three images: (a) Beach wood, (b) TP_C01_023, and (c) Japan tower.

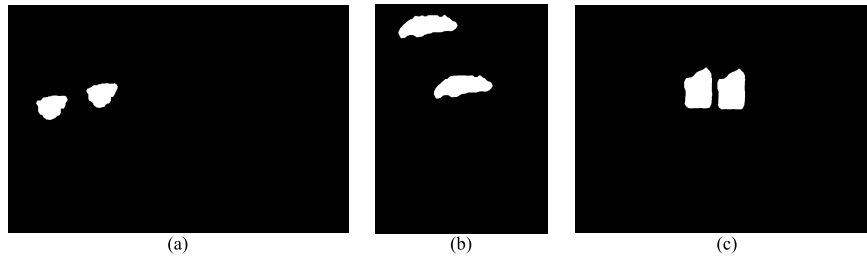


FIGURE 10. Final detection results of the proposed method on three images: (a) Beach wood, (b) TP_C01_023, and (c) Japan tower.

Algorithm 1 RANSAC Algorithm

Variable Declaration:

- x_m : matched points by the improved g2NN algorithm
- x_{in} : matched points belonging to inlier group via the RANSAC algorithm
- N_{in} : number of the points belonging in inlier group
- T_{in} : threshold of the points belonging to inlier group
- x'_m : matched points after eliminating false matched points by RANSAC

RANSAC Iteration Procedure:

```

while (1)
     $x_{in} = \text{RANSAC}(x_m)$ 
    if  $N_{in} > T_{in}$  then
         $x'_m \leftarrow x_{in}$ 
         $x_{in}$  is eliminated from  $x_m$ 
    else
        break
    end if
end while
End Procedure
    
```

$x_{min}^i - N_p, x_{max}^i + N_p, y_{min}^i - N_p,$ and $y_{max}^i + N_p,$ considering $x_{min}^i, y_{min}^i, x_{max}^i,$ and y_{max}^i may not cover the whole actual tampered region [46], where the N_p is the number of extended edge pixels. After obtaining the rectangle regions, these regions are divided into overlapping circular blocks of radius R_{PCET} pixels, and the PCET coefficients are extracted from each circle block. Similar to Section IV.C, the improved g2NN algorithm is used to find similar features. Then, mathematical morphology close and open operations are used

to eliminate isolated small regions and fill in holes. The mask I_b , a binary image where 0 indicates black background and 1 indicates the detected regions, is obtained. Finally, an iterative strategy is used to expand the detected tampered regions. For each (x, y) in I_b , if 0 exists in its 8 neighborhood regions and $I_b(x, y) = 1$, the corresponding regions are set to 1. Three examples of the final detected regions are given in Fig. 10.

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

This section introduces the evaluation metric first. And then the plain tests, the geometric transformations tests, and robustness tests are performed to evaluate the performance of the proposed method.

The tests in the paper are performed in MATLAB 2019a on a 64-bit win10 PC with the Intel Core i5-4690 CPU model and 8 GB RAM.

In state-of-the-art methods, Cozzolino *et al.* [26] and Zandi [39] specifically recommended that their methods can locate the tampered region in smooth regions. Pun *et al.* [40] integrated block-based and keypoint-based CMFD methods, which is similar to the proposed method. Therefore, the proposed method is compared with the methods [26], [39], and [40] to evaluate its performance.

The images used in the experiments are from three datasets: GRIP [26], FAU [45], and SBU-CM16 [47] datasets, which are also used in the methods [26], [39], and [40]. The GRIP dataset [26], created by Cozzolino *et al.*, includes 80 original images, plain copy-move forgery images, and corresponding binary images of the identified tampered regions. The FAU dataset [45], created by Christlein *et al.*, includes 48 groups of images. Each group contains plain forgery images and

distorted copy-move forgery images such as rotation, scaling, blurring, JPEG compression, and noise addition. The original images and binary images of the tampered region corresponding to the forgery images are also given. The SBU-CM16 dataset [47], created by Zandi *et al.*, includes 16 groups of images. Each group contains forgery images and binary images after rotation, blurring, JPEG compression, and noise addition, excluding the plain forgery images and original images.

In CMFD methods, the precision p , recall r , and F score metrics are commonly used to evaluate the performance of method and are defined as follows [45]:

$$p = \frac{N_{cf}}{N_{cf} + N_{ff}}, \quad r = \frac{N_{cf}}{N_{cf} + N_{fo}}, \quad F = \frac{2pr}{p + r}, \quad (15)$$

where N_{cf} is the number of pixels correctly detected as forged, N_{ff} is the number of pixels falsely detected as forged, and N_{fo} is the number of pixels falsely detected as original.

In (15), p , r , $F \in [0, 1]$. The smaller the value of p , the smaller the area of the detected region than the real tampered region. The smaller the value of r , the larger the area of the detected region than the real tampered region. The closer p , r , and F are to 1, the higher accuracy the CMFD method in detecting tampered region has.

The parameters used in the proposed method are listed in Table 3.

TABLE 3. Parameters setting in the proposed method.

| Parameter | Value | Meaning |
|--------------|-------|------------------------------------------------------------------|
| T_{SURF}^s | 0 | Threshold for selecting the keypoints of SURF in smooth regions |
| T_{SURF}^t | 0.01 | Threshold for selecting the keypoints of SURF in texture regions |
| R_{PCET} | 8 | Radius of image block for obtaining PCET values |
| L | 3 | Limitation number in PHT |
| T_{g2NN} | 0.6 | Ratio threshold in the improved g2NN algorithm |
| T_d | 100 | Distance threshold of the matched keypoints |
| N_{KNN} | 10 | Number of nearest points in KNN |
| T_{in} | 10 | Threshold of points in inlier group |
| T_L | 10 | Threshold of corresponding label frequency |
| N_p | 80 | Number of extended edge pixels |

A. PLAIN IMAGE CMFD TESTS

In the tests of plain copy-move forgery images, the representative images shown in Fig. 11 are selected from the GRIP [26] and FAU [45] datasets to demonstrate the effectiveness of the proposed method. The forgery of TP_C01_023 shown in Fig. 11(a) occurs in smooth region. The forgeries of TP_C01_005 and TP_C01_030 shown in Fig. 11(b) and Fig. 11(c) occur in high-brightness smooth regions. The forgeries of TP_C01_024 and TP_C01_049 shown in Fig. 11(d) and Fig. 11(e) occur in forgery images involving too many similar but genuine

regions. The forgery of Bricks shown in Fig. 11(f) occurs in multiple regions.

The detection results of [26], [39], [40], and the proposed method are shown in Fig. 11. In Fig. 11, the images in the first to seventh rows are: original images, forgery images, ground-truth tampered regions, detection results of the PCT-cart method [26], iterative method [39], SLIC method [40], and detection results of the proposed method based on SURF and PCET.

For the forgery which occurs in smooth region of the sky, as shown in Fig. 11(a), the located effects of [26], [39], and the proposed method are better than those of [40]. The method [40] has a bad edge processing, resulting in the smaller detected region. Since the iterative strategy makes the edge of the detected region slightly smooth, the detection results of the proposed method are a little worse than those of [26]. For the forgery which occurs in high-brightness smooth regions, as shown in Fig. 11(b) and Fig. 11(c), the methods [26] and [40] cannot locate the tampered regions of the forgery image. The reason for this is that the tampered regions are small and high-brightness, leading to that only a few keypoints are detected in the tampered regions. In addition, the detection results of [39] regard the original pixels as forged due to the existence of false matched points. For the forgery which occurs in forgery images involving too many similar but genuine regions, as shown in Fig. 11(d) and Fig. 11(e), the detection results of [26] and [39] regard the original pixels as forged and the location of [40] is smaller than the real tampered regions. Since the filtering strategy combines the label matrix to eliminate false matched points, the proposed method avoids the problems and locates the tampered regions more accurately. For multiple copy-move forgeries, multiple tampered regions have different affine transformations, as shown in Fig. 11(f). The proposed method can locate the tampered regions accurately, which is similar to the detection results of [26] and [39], and better than those of [40].

In addition, the precision p , recall r , and F score of the detection results in Fig. 11 are calculated respectively and listed in Table 4. Table 4 shows that the proposed method is better than [26], [39], and [40] in locating the tampered regions. The F score of the proposed method is significantly higher than that of [26], [39], and [40]. It indicates that the proposed method is superior to [26], [39], and [40] in the location of the plain copy-move forgery images, especially when the forgery occurs in high-brightness smooth regions or forgery images involving similar but genuine regions.

B. GEOMETRIC TRANSFORMATION TESTS

The above tests are performed only on plain copy-move forgery images. However, image forgery usually performs geometric transformations before plain copy-move forgery, such as rotation and scaling. Therefore, the rotation and scaling tests are necessary to evaluate the performance of the proposed method.

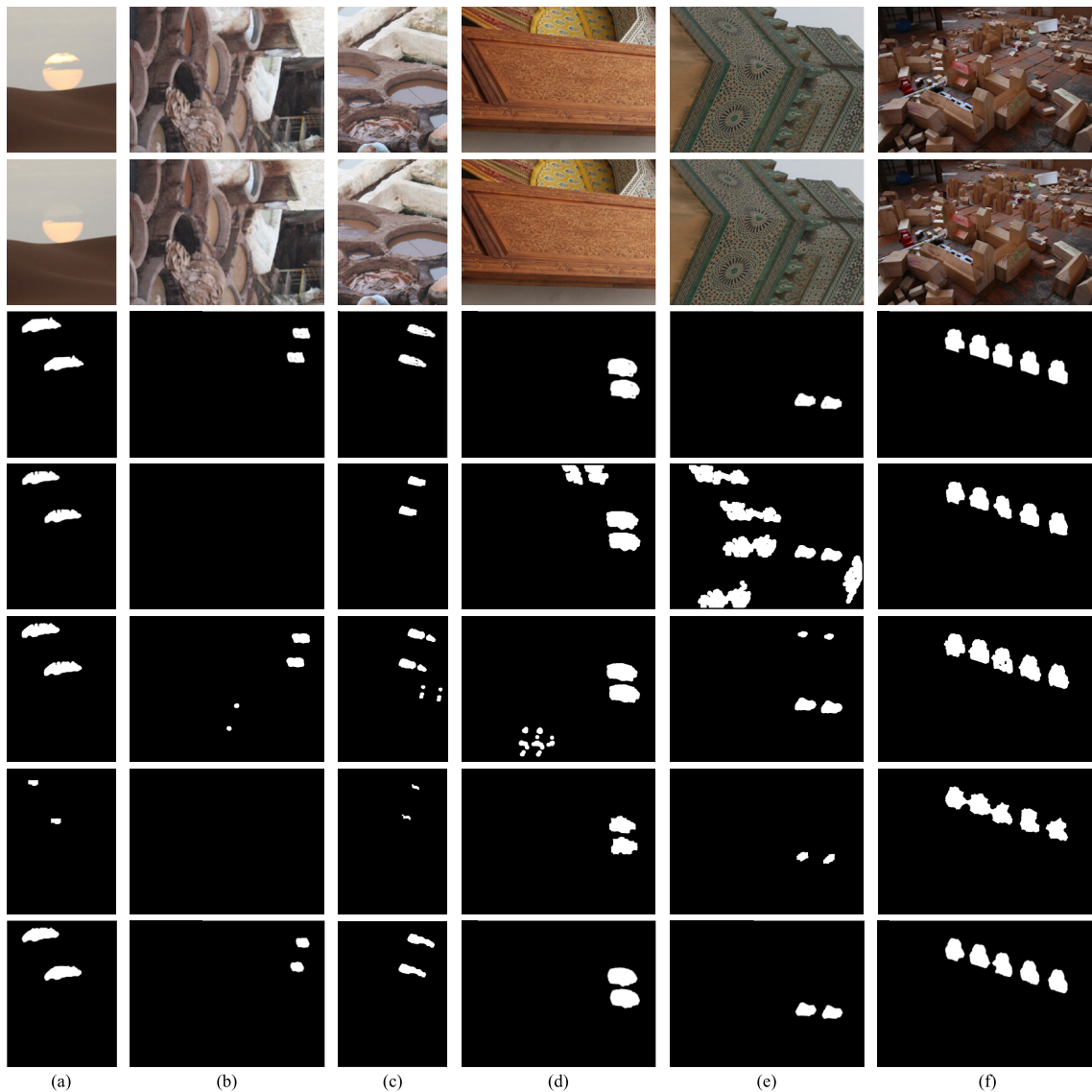


FIGURE 11. Detection results of plain image copy-move forgery on six images. First row: original images; Second row: forgery images; Third row: ground-truth tampered regions; Fourth row: detection results of the PCT-cart method [26]; Fifth row: detection results of the iterative method [39]; Sixth row: detection results of the SLIC method [40]; Seventh row: detection results of the proposed method based on SURF and PCET. (a) TP_C01_023, (b) TP_C01_005, (c) TP_C01_030, (d) TP_C01_024, (e) TP_C01_049, and (f) Bricks.

TABLE 4. The precision p , recall r , and F score of the detected images shown in Figure 11.

| Method | | [26] | | | [39] | | | [40] | | | Proposed | | |
|-------------------------------|------------|------|------|-------------|------|------|------|------|------|------|----------|------|-------------|
| Metric | | p | r | F | p | r | F | p | r | F | p | r | F |
| Smooth region | TP_C01_023 | 1.00 | 0.95 | 0.97 | 0.98 | 0.96 | 0.97 | 1.00 | 0.13 | 0.23 | 0.97 | 0.95 | 0.96 |
| High-brightness smooth region | TP_C01_005 | 0.00 | 0.00 | 0.00 | 0.85 | 0.92 | 0.88 | 0.00 | 0.00 | 0.00 | 0.97 | 0.91 | 0.94 |
| | TP_C01_030 | 0.99 | 0.70 | 0.82 | 0.79 | 0.86 | 0.82 | 1.00 | 0.12 | 0.21 | 0.94 | 0.90 | 0.92 |
| Similar region | TP_C01_024 | 0.57 | 1.00 | 0.72 | 0.73 | 0.98 | 0.83 | 0.99 | 0.75 | 0.85 | 0.97 | 0.97 | 0.97 |
| | TP_C01_049 | 0.09 | 1.00 | 0.16 | 0.79 | 0.98 | 0.87 | 0.97 | 0.37 | 0.54 | 0.95 | 0.96 | 0.96 |
| Multiple regions | Bricks | 0.99 | 0.95 | 0.97 | 0.91 | 0.99 | 0.95 | 0.90 | 0.88 | 0.89 | 0.99 | 0.94 | 0.96 |

In the rotation tests, the copied region is rotated by an angle before pasting it into another region in the corresponding image. The rotation tests include slight rotation

tests and large rotation tests. The F scores of detection results of [26], [39], [40], and the proposed method in slight rotation tests and large rotation tests are listed

TABLE 5. The F scores of the detection results on five images in slight rotation tests.






| Rotated image |  |  |  |  |  | Mean |
|----------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|-------------|
| Rotation angle | 0° | 2° | 4° | 6° | 8° | |
| [26] | 0.82 | 1.00 | 0.62 | 0.99 | 0.99 | 0.88 |
| [39] | 0.82 | 0.74 | 0.70 | 0.54 | 0.84 | 0.73 |
| [40] | 0.21 | 0.92 | 0.25 | 0.07 | 0.71 | 0.43 |
| Proposed | 0.92 | 0.95 | 0.81 | 0.94 | 0.96 | 0.92 |

TABLE 6. The F scores of the detection results on five images in large rotation tests.






| Rotated image |  |  |  |  |  | Mean |
|----------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------|
| Rotation angle | 10° | 30° | 50° | 70° | 90° | |
| [26] | 0.98 | 0.95 | 0.97 | 0.97 | 0.97 | 0.97 |
| [39] | 0.98 | 0.91 | 0.96 | 0.99 | 0.98 | 0.96 |
| [40] | 0.89 | 0.45 | 0.74 | 0.46 | 0.68 | 0.64 |
| Proposed | 0.98 | 0.96 | 0.97 | 0.95 | 0.93 | 0.96 |

TABLE 7. The F scores of the detection results on five images in scale down tests.






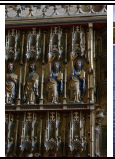




| Scaled image |  |  |  |  |  | Mean |
|----------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|-------------|
| Scaling factor | 0.91 | 0.93 | 0.95 | 0.97 | 0.99 | |
| [26] | 0.61 | 0.99 | 0.99 | 0.88 | 0.94 | 0.88 |
| [39] | 0.71 | 0.96 | 0.74 | 0.85 | 0.68 | 0.79 |
| [40] | 0.22 | 0.62 | 0.00 | 0.77 | 0.35 | 0.39 |
| Proposed | 0.77 | 0.96 | 0.91 | 0.96 | 0.95 | 0.91 |

TABLE 8. The F scores of the detection results on five images in scale up tests.

| Scaled image |  |  |  |  |  | Mean |
|----------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|-------------|
| Scaling factor | 1.01 | 1.03 | 1.05 | 1.07 | 1.09 | |
| [26] | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 0.99 |
| [39] | 0.77 | 0.69 | 0.82 | 0.96 | 0.00 | 0.65 |
| [40] | 0.60 | 0.76 | 0.00 | 0.80 | 0.93 | 0.62 |
| Proposed | 0.97 | 0.95 | 0.91 | 0.92 | 0.96 | 0.94 |

in Table 5 and Table 6, respectively. In Table 5, the tampered regions of the images are slightly rotated by 0°, 2°, 4°, 6°, and 8°, respectively. In Table 6, the tampered regions of the images are large-scale rotated by 10°, 30°, 50°, 70°, and 90°, respectively.

In the scaling tests, the copied region is scaled by a factor before pasting it into another area in the same image. The scaling tests include scale down tests and scale up tests. The F scores of detection results of [26], [39], [40], and the proposed method in scale down tests and scale up tests are

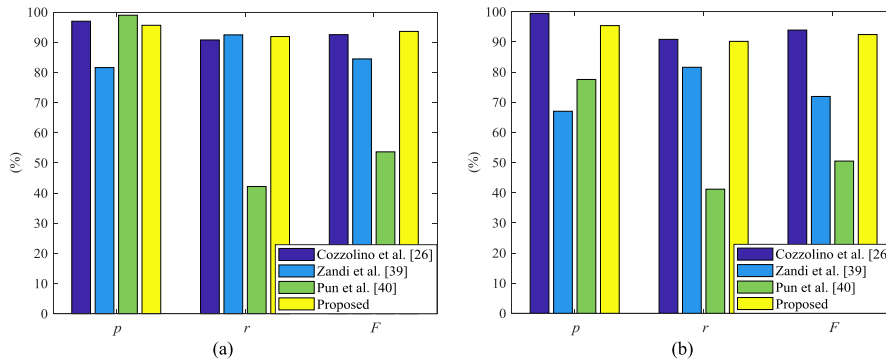


FIGURE 12. The average values of p , r , and F of the proposed method and other CMFD methods under two geometric transformations: (a) rotation and (b) scaling.

listed in Table 7 and Table 8, respectively. In Table 7, the tampered regions of the images are scaled down by 0.91, 0.93, 0.95, 0.97, and 0.99, respectively. In Table 8, the tampered regions of the images are scaled up by 1.01, 1.03, 1.05, 1.07, and 1.09, respectively.

Table 5, Table 6, Table 7, and Table 8 show that the proposed method is similar or better than other methods [26], [39], and [40]. To evaluate the performance of the proposed method comprehensively, the average values of p , r , and F of the detection results using [26], [39], [40], and the proposed method, respectively, are calculated and shown in Fig. 12. The data in Fig. 12(a) are obtained by averaging the detection results on total ten images in Table 5 and Table 6. The data in Fig. 12(b) are obtained by averaging the detection results on total ten images in Table 7 and Table 8. In the rotation tests and the scaling tests, the proposed method is superior to [39] and [40], and similar to [26], shown in Fig. 12. The F score of the proposed method is slightly lower than that of [26] due to the smooth edge. In addition, the reason why the F score of [40] is much lower is that the method cannot locate some tampered regions because the tampered regions are divided into many image blocks by superpixel segmentation.

C. ROBUSTNESS TESTS

After copy-move forgery, some different post-processing manipulations are usually used to cover tampered traces, such as blurring, JPEG compression, and noise addition. To cover tampered traces, these manipulations could weaken or cover the edges caused by image forgery, which means the CMFD method requires the ability to resist these manipulations.

The F scores of detection results in robustness tests including blurring tests, JPEG compression tests, and noise addition tests are listed in Table 9, Table 10, and Table 11, respectively. In the blurring tests, to obtain blurred images shown in Table 9, the images are filtered by circular averaging filters. The filter is the square matrix of size $2r_c + 1$, where r_c is a radius of circular filter ranging from 0.4 to 2.5. In the JPEG compression tests, the images are JPEG compressed with different QFs to obtain JPEG format images, as shown

in Table 10. The QF ranges from 100 to 50 with a step size of 10. The QF denotes the compression degree. The higher the QF, the better the image quality. In the noise addition tests, Gaussian noise with zero mean and different variances are added to the images, as shown in Table 11. In addition, several images in the robustness tests are selected from the SBU-CM16 [47] directly, that's why the filter sizes in the blurring tests have different intervals.

The tampered regions of these blurred images, JPEG compressed images, and noise added images are detected by [26], [39], [40], and the proposed method, respectively. Table 9, Table 10, and Table 11 show that the proposed method is better than other methods [26], [39], and [40]. To evaluate the performance of the proposed method comprehensively, the average values of p , r , and F of the detection results using [26], [39], [40], and the proposed method, respectively, are calculated and shown in Fig. 13. The data in Fig. 13(a) are obtained by averaging the detection results on total six images in Table 9. The data in Fig. 13(b) are obtained by averaging the detection results on total six images in Table 10. The data in Fig. 13(c) are obtained by averaging the detection results on total five images in Table 10. The detection results display that the proposed method can detect the tampered regions of the forgery images even if they are subjected to varying degrees of blurring, JPEG compression, and noise addition. As shown in Fig. 13, the proposed method is superior to [26], [39], and [40] in terms of resisting blurring, JPEG compression, and noise addition. In addition, in the detection results of noise addition tests, it can be seen that the noise interference is enormous when the forgery occurs in forgery images involving similar but genuine regions.

In summary, the proposed CMFD method can detect tampered regions in plain and multiple forgeries in images, and can resist different distortions by various attacks, including rotation, scaling, blurring, JPEG compression, and noise addition. Compared with other CMFD methods [26], [39], and [40], the proposed method is more accurate in terms of detecting forgery, especially when the forgery occurs in high-brightness smooth regions or forgery images involving similar but genuine regions. In terms of resisting rotation,

TABLE 9. The F scores of the detection results on six images in blurring tests.






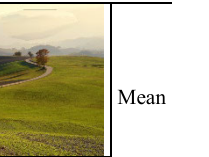
| | | | | | | | |
|---------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------|
| Blurred image |  |  |  |  |  |  | Mean |
| Filter size | 0.4 | 0.5 | 1.2 | 1.5 | 2.0 | 2.5 | |
| [26] | 0.64 | 0.95 | 0.96 | 0.97 | 0.36 | 0.95 | 0.80 |
| [39] | 0.83 | 0.82 | 0.96 | 0.58 | 0.89 | 0.79 | 0.81 |
| [40] | 0.85 | 0.00 | 0.46 | 0.72 | 0.71 | 0.00 | 0.46 |
| Proposed | 0.97 | 0.97 | 0.96 | 0.96 | 0.96 | 0.90 | 0.95 |

TABLE 10. The F scores of the detection results on six images in JPEG compression tests.

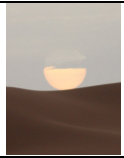
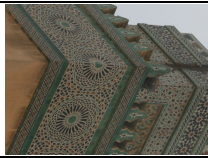









| | | | | | | | |
|-----------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------|
| JPEG compressed image |  |  |  |  |  |  | Mean |
| QF | 100 | 90 | 80 | 70 | 60 | 50 | |
| [26] | 0.97 | 0.15 | 0.94 | 0.97 | 0.64 | 0.99 | 0.78 |
| [39] | 0.97 | 0.87 | 0.92 | 0.71 | 0.97 | 0.97 | 0.90 |
| [40] | 0.00 | 0.00 | 0.00 | 0.74 | 0.32 | 0.71 | 0.29 |
| Proposed | 0.96 | 0.96 | 0.96 | 0.97 | 0.96 | 0.97 | 0.96 |

TABLE 11. The F scores of the detection results on five images in noise addition tests.

| | | | | | | |
|----------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|-------------|
| Noised image |  |  |  |  |  | Mean |
| Noise variance | 0.001 | 0.001 | 0.001 | 0.002 | 0.003 | |
| [26] | 0.68 | 0.15 | 0.95 | 0.97 | 0.95 | 0.74 |
| [39] | 0.96 | 0.00 | 0.85 | 0.79 | 0.63 | 0.65 |
| [40] | 0.25 | 0.00 | 0.00 | 0.69 | 0.05 | 0.20 |
| Proposed | 0.89 | 0.92 | 0.93 | 0.97 | 0.85 | 0.91 |

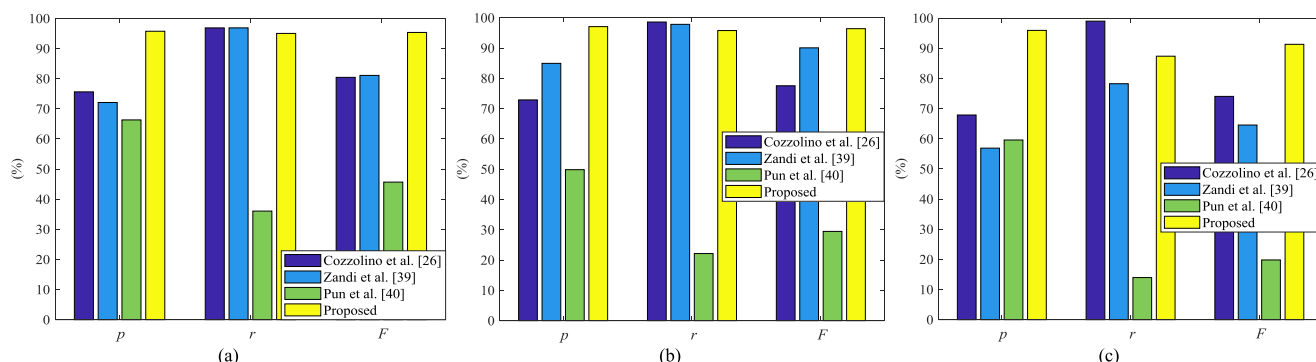


FIGURE 13. The average values of p , r , and F of the proposed method and other CMFD methods under three post-processing manipulations: (a) blurring, (b) JPEG compression, and (c) noise addition.

scaling, blurring, JPEG compression, and noise addition, the proposed CMFD method is more effective than [26], [39], and [40]. However, the cost of these advantages is the running

time related to the size of the tampered region, which means the running time will be long when the tampered region becomes large.

VI. CONCLUSION

Aiming at the difficulty of detecting forgery which occurs in high-brightness smooth regions or forgery images involving similar but genuine regions correctly, an image CMFD method based on SURF and PCET is proposed. The proposed method combines the advantages of block-based and keypoint-based image CMFD methods. Since features are extracted and matched based on keypoints, the proposed method has low computational complexity and accurate detection in tampered regions. In addition, false matched points are eliminated based on blocks, which makes the proposed method has low probability of false matching. The experiments have proved that the proposed method based on SURF and PCET can locate the tampered regions of the copy-move forgery image in the high-brightness smooth regions. Moreover, the false matching situation can be avoided, when the forgery occurs in forgery images involving similar but genuine regions. The proposed method based on SURF and PCET can resist different distortions by various attacks, including rotation, scaling, blurring, JPEG compression, and noise addition.

However, there are several aspects need to be improved in the future. In mathematical morphology operations and hierarchical cluster sections, the parameters are difficult to generalize in various conditions. In feature matching section, a new time-saving method is desired to be created. In addition, due to the sampling or interpolation of large-scale reduction or enlargement in the image regions, it is difficult to detect whether the images are tampered using the existing methods.

REFERENCES

- [1] K. Asghar, Z. Habib, and M. Hussain, "Copy-move and splicing image forgery detection and localization techniques: A review," *Austral. J. Forensic Sci.*, vol. 49, no. 3, pp. 281–307, 2017.
- [2] *Photoshop*. Accessed: Nov. 20, 2019. [Online]. Available: <https://www.photoshop.com/>
- [3] *ACDSee*. Accessed: Nov. 20, 2019. [Online]. Available: <https://www.acdsee.com/>
- [4] L. Zheng, Y. Zhang, and L. Vrizlynn, "A survey on image tampering and its detection in real-world photos," *J. Vis. Commun. Image Represent.*, vol. 58, pp. 380–399, Jan. 2019.
- [5] S. Sharma and U. Ghanekar, "A rotationally invariant texture descriptor to detect copy move forgery in medical images," in *Proc. IEEE Int. Conf. Comput. Intell. Commun. Technol.*, Ghaziabad, India, Feb. 2015, pp. 795–798.
- [6] *Photo Tampering Throughout History*. Accessed: Nov. 20, 2019. [Online]. Available: https://pth.izitru.com/2016_02_01.html
- [7] X. Lin, J.-H. Li, S.-L. Wang, A.-W.-C. Liew, F. Cheng, and X.-S. Huang, "Recent advances in passive digital image security forensics: A brief review," *Engineering*, vol. 4, pp. 29–39, Feb. 2018.
- [8] C. Wang, H. Zhang, and X. Zhou, "A self-recovery fragile image watermarking with variable watermark capacity," *Appl. Sci.*, vol. 8, no. 4, Apr. 2018, Art. no. 548.
- [9] A. Zear, A. K. Singh, and P. Kumar, "A proposed secure multiple watermarking technique based on DWT, DCT and SVD for application in medicine," *Multimedia Tools Appl.*, vol. 77, no. 4, pp. 4863–4882, Feb. 2018.
- [10] A. Shehab, M. Elhoseny, K. Muhammad, A. K. Sangaiyah, P. Yang, H. Huang, and G. Hou, "Secure and robust fragile watermarking scheme for medical images," *IEEE Access*, vol. 6, pp. 10269–10278, Feb. 2018.
- [11] X. Wang, J. Xue, Z. Zheng, Z. Liu, and N. Li, "Image forensic signature for content authenticity analysis," *J. Vis. Commun. Image Represent.*, vol. 23, no. 5, pp. 782–797, Jul. 2012.
- [12] M. Okawa, "From BoVW to VLAD with KAZE features: Offline signature verification considering cognitive processes of forensic experts," *Pattern Recognit. Lett.*, vol. 113, pp. 75–82, Oct. 2018.
- [13] M. Okawa, "Synergy of foreground–background images for feature extraction: Offline signature verification using Fisher vector with fused KAZE features," *Pattern Recognit.*, vol. 79, pp. 480–489, Jul. 2018.
- [14] S. Teerakanok and T. Uehara, "Copy-move forgery detection: A state-of-the-art technical review and analysis," *IEEE Access*, vol. 7, pp. 40550–40568, 2019.
- [15] A. J. Fridrich, D. Soukal, and J. Lukáš, "Detection of copy-move forgery in digital images," in *Proc. Digit. Forensic Res. Workshop*, Cleveland, OH, USA, Aug. 2003, pp. 55–61.
- [16] Y. Huang, W. Lu, W. Sun, and D. Long, "Improved DCT-based detection of copy-move forgery in images," *Forensic Sci. Int.*, vol. 206, nos. 1–3, pp. 178–184, Mar. 2011.
- [17] J. Zhao and J. Guo, "Passive forensics for copy-move image forgery using a method based on DCT and SVD," *Forensic Sci. Int.*, vol. 233, nos. 1–3, pp. 158–166, Dec. 2013.
- [18] G. Li, Q. Wu, D. Tu, and S. Sun, "A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD," in *Proc. IEEE Int. Conf. Multimedia Expo*, Beijing, China, Jul. 2007, pp. 1750–1753.
- [19] W. Luo, J. Huang, and G. Qiu, "Robust detection of region-duplication forgery in digital image," in *Proc. Int. Conf. Pattern Recognit.*, Hong Kong, Aug. 2006, pp. 746–749.
- [20] G. Liu, J. Wang, S. Lian, and Z. Wang, "A passive image authentication scheme for detecting region-duplication forgery with rotation," *J. Netw. Comput. Appl.*, vol. 34, no. 5, pp. 1557–1565, Sep. 2011.
- [21] J. Ouyang, Y. Liu, and M. Liao, "Robust copy-move forgery detection method using pyramid model and Zernike moments," *Multimedia Tools Appl.*, vol. 78, no. 8, pp. 10207–10225, Apr. 2019.
- [22] B. Soni, P. K. Das, and D. M. Thounaojam, "Dual system for copy-move forgery detection using block-based LBP-HF and FWHT features," *Eng. Lett.*, vol. 26, no. 1, pp. 171–180, Feb. 2018.
- [23] J. Deng, J. Yang, S. Weng, G. Gu, and Z. Li, "Copy-move forgery detection robust to various transformation and degradation attacks," *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 9, pp. 4467–4486, Sep. 2018.
- [24] P.-T. Yap, X. Jiang, and A. C. Kot, "Two-dimensional polar harmonic transforms for invariant image representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1259–1270, Jul. 2010.
- [25] K. M. Hosny, H. M. Hamza, and N. A. Lashin, "Copy-move forgery detection of duplicated objects using accurate PCET moments and morphological operators," *Imag. Sci. J.*, vol. 66, no. 6, pp. 330–345, Aug. 2018.
- [26] D. Cozzolino, G. Poggi, and L. Verdoliva, "Efficient dense-field copy-move forgery detection," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 11, pp. 2284–2297, Nov. 2015.
- [27] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, "A SIFT-based forensic method for copy-move attack detection and transformation recovery," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 1099–1110, Sep. 2011.
- [28] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, L. Del Tongo, and G. Serra, "Copy-move forgery detection and localization by means of robust clustering with J-Linkage," *Signal Process., Image Commun.*, vol. 28, no. 6, pp. 659–669, Jul. 2013.
- [29] P. Mishra, N. Mishra, S. Sharma, and R. Patel, "Region duplication forgery detection technique based on SURF and HAC," *Sci. World J.*, vol. 2013, Sep. 2013, Art. no. 267691.
- [30] L. Chen, W. Lu, J. Ni, W. Sun, and J. Huang, "Region duplication detection based on Harris corner points and step sector statistics," *J. Vis. Commun. Image Represent.*, vol. 24, no. 3, pp. 244–254, Apr. 2013.
- [31] Y. Liu, H.-X. Wang, H.-Z. Wu, and Y. Chen, "An efficient copy-move detection algorithm based on superpixel segmentation and Harris keypoints," in *Proc. 3rd Int. Conf. Cloud Comput. Secur.*, in Lecture Notes in Computer Science, Nanjing, China, vol. 10602, Jun. 2017, pp. 61–73.
- [32] G. Ulutas and G. Muzaffer, "A new copy move forgery detection method resistant to object removal with uniform background forgery," *Math. Problems Eng.*, vol. 2016, Oct. 2016, Art. no. 3215162.
- [33] Y. Zhu, X. Shen, and H. Chen, "Copy-move forgery detection based on scaled ORB," *Multimedia Tools Appl.*, vol. 75, no. 6, pp. 3221–3233, Mar. 2016.
- [34] L. Yu, Q. Han, and X. Niu, "Feature point-based copy-move forgery detection: Covering the non-textured areas," *Multimedia Tools Appl.*, vol. 75, no. 2, pp. 1159–1176, Jan. 2016.

- [35] M. M. Isaac and M. Wilsy, "Copy-move forgery detection based on Harris corner points and BRISK," in *Proc. 3rd Int. Symp. Women Comput. Inform.*, Kochi, India, Aug. 2015, pp. 394–399.
- [36] X. Pan and S. Lyu, "Region duplication detection using image feature matching," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 857–867, Dec. 2010.
- [37] F. Yang, J. Li, W. Lu, and J. Weng, "Copy-move forgery detection based on hybrid features," *Eng. Appl. Artif. Intell.*, vol. 59, pp. 73–83, Mar. 2017.
- [38] G. Jin and X. Wan, "An improved method for SIFT-based copy-move forgery detection using non-maximum value suppression and optimized J-Linkage," *Signal Process., Image Commun.*, vol. 57, pp. 113–125, Sep. 2017.
- [39] M. Zandi, A. Mahmoudi-Aznavah, and A. Talebpour, "Iterative copy-move forgery detection based on a new interest point detector," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 11, pp. 2499–2512, Nov. 2016.
- [40] C. Pun, X. Yuan, and X. Bi, "Image forgery detection using adaptive oversegmentation and feature point matching," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 8, pp. 1705–1716, Aug. 2015.
- [41] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, Abu Dhabi, United Arab Emirates, Dec. 2016, pp. 1–6.
- [42] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.
- [43] Y. Hu, Y. Li, R. Song, P. Rao, and Y. Wang, "Minimum barrier superpixel segmentation," *Image Vis. Comput.*, vol. 70, pp. 1–10, Feb. 2018.
- [44] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2011, pp. 2097–2104.
- [45] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou, "An evaluation of popular copy-move forgery detection approaches," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 6, pp. 1841–1854, Dec. 2012.
- [46] C. Wang, Z. Zhang, and X. Zhou, "An image copy-move forgery detection scheme based on A-KAZE and SURF features," *Symmetry*, vol. 10, no. 12, Dec. 2018, Art. no. 706.
- [47] M. Zandi, A. Mahmoudi-Aznavah, and A. Mansouri, "Adaptive matching for copy-move forgery detection," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, Atlanta, GA, USA, Dec. 2014, pp. 119–124.



vision, machine learning, and wireless communication technology.

CHENGYOU WANG (M'16) received the B.E. degree in electronic information science and technology from Yantai University, China, in 2004, and the M.E. and Ph.D. degrees in signal and information processing from Tianjin University, China, in 2007 and 2010, respectively. He is currently an Associate Professor and a Supervisor of master's students with Shandong University, Weihai, China. His current research interests include digital image/video processing and analysis, computer



ZHI ZHANG received the B.E. degree in electronic information engineering from the Shandong University of Science and Technology, China, in 2016, and the M.E. degree in information and communication engineering from Shandong University, China, in 2019. His current research interests include image forgery detection and computer vision.



QIANWEN LI received the B.E. degree in electronic information engineering from the Shandong University of Science and Technology, China, in 2019. She is currently pursuing the M.E. degree in information and communication engineering with Shandong University, China. Her current research interests include image forgery detection, image watermarking, and computer vision.



Her current research interests include wireless communication technology, digital image processing, and computer vision.

XIAO ZHOU (M'19) received the B.E. degree in automation from the Nanjing University of Posts and Telecommunications, China, in 2003, the M.E. degree in information and communication engineering from Inha University, South Korea, in 2005, and the Ph.D. degree in information and communication engineering from Tsinghua University, China, in 2013. She is currently an Associate Professor and a Supervisor of master's students with Shandong University, Weihai, China.

...