IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 23, 2020, accepted February 1, 2020, date of publication February 18, 2020, date of current version March 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2974804

Copy-Move Forgery Detection Based on Keypoint Clustering and Similar Neighborhood Search Algorithm

HAIPENG CHEN[1,2], XIWEN YANG[2,3], AND YINGDA LYU[4]

[1]College of Computer Science and Technology, Jilin University, Changchun 130012, China
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun 130012, China
[3]College of Software, Jilin University, Changchun 130012, China
[4]Public Computer Education and Research Center, Jilin University, Changchun 130012, China

Corresponding author: Yingda Lyu (ydlv@jlu.edu.cn)

This work was supported in part by the Jilin Province Science and Technology Development Plan Project under Grant 20190303134SF and Grant 20180201064SF, in part by the National Natural Science Foundation of China under Grant 61672259, Grant 61876070, and Grant 61602203, and in part by the National Science Foundation of Jilin Province under Grant 20180520020JH.

ABSTRACT Copy-move is one of the most commonly used methods of tampering with digital images. Keypoint-based detection is recognized as effective in copy-move forgery detection (CMFD). This paper proposes an efficient CMFD method via clustering SIFT keypoints and searching the similar neighborhoods to locate tampered regions. In the proposed method, the keypoints are clustered based on scale and color, grouped into several smaller clusters and matched separately, which reduce the high time complexity caused in matching caused by the high dimensionality of SIFT. In order to locate the tampered regions accurately at pixel level finally, a novel localization algorithm is designed to compare the similar neighborhoods of matching pairs by two similarity measures, and mark the tampered regions in pixels iteratively. We experimented on three different image data sets including kinds of tampering means to compare and verify the effectiveness and robustness of proposed method. The experimental results show that the proposed method is superior to existing state-of-art methods in terms of matching time complexity, detection reliability and forgery location accuracy.

INDEX TERMS Copy-move forgery detection, digital image forensics, keypoint clustering, similar neighborhood search algorithm.

I. INTRODUCTION

Multimedia image information is often used as evidence in many important occasions, such as criminal investigations and military scenarios. However, with the development of technology and network, digital images can be easily acquired and tampered with, which makes the authenticity of digital images face serious risks and poses a great threat to Judicial Forensics and various research work. Among them, copy-move forgery is one of the most common means of image forgery, which by copying certain regions of the image and pasting them into elsewhere in the same image. This makes the attacker tamper digit images easily through utilizing the same illumination angle, imaging equipment and other characteristics in the same image, in order to hide or emphasize certain objects. An example of digital image tampering is shown in the right image of Fig. 1 that was published on the Iranian revolutionary guard website in which four missiles appeared to take off from a desert launch pad. However, analysts reported that three missiles were actually launched as in the left image of Fig.1. In the tampered image, the marked regions of the image appear to be closely replicated [1].

FIGURE 1. Example of image forgery: original image(left) and tampered image (right).

In recent years, the authenticity and dependability of images have become the hot issues of research, and a

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

key technology in digital image identification is copy-move forgery detection (CMFD). The goal of CMFD is to find some regions similar to other regions of the image, because the tampered part is copied from the same image. Meanwhile, in the process of tampering, geometric or post-processing operations are usually preformed onto tampered regions in order to make the forgery real and unnoticeable. The strong similarity between the tampered regions and the source regions has become the important evidence in CMFD.

However, the time complexity of existing methods is high, especially in the feature matching stage, and the location of tampered regions is not accurate enough to meet the practical requirements. Since in practical forensics applications, figuring out the tampered regions compared to forgery detections is more important and necessary [33]. In this paper, an improved CMFD method is proposed, which includes clustering the keypoints before matching and locating the tampered regions at pixel level. Our main contributions can be summarized as:

- In order to solve the problem of high matching time complexity, keypoints clustering algorithm based on scale and color is proposed. Using the high similarity between the keypoints in tampered regions and source regions, we grouped the keypoints based on scale and color respectively. The keypoints in each group are matched separately, which can greatly shorten the matching time.
- An algorithm for locating tampered regions by searching similar neighborhoods iteratively at the pixel level is proposed, which introduces two similarity evaluation metrics, Polar Cosine Transform (PCT) features and Peak Signal-to-Noise Ratio (PSNR), to figure out tampered regions more accurately.
- We have experimented on three datasets for different types of forgery attacks. The experimental results show that this method can effectively overcome the shortcomings of traditional methods, such as large computation and time-consuming, and can locate tampered regions efficiently and robustly.

The remainder of this paper is organized as follows: Section II gives a brief introduction of the CMFD methods. An overview about SIFT feature is presented in Section III. In Section IV, the method framework is described and its details are introduced briefly. Section V assesses our proposed method through a series of experiments. Finally, the conclusion is provided in Section VI.

II. RELATED WORK

In this section, we discuss the research status and describe the methods involved in CMFD. Specifically, the techniques can be divided into block-based, keypoint-based and other fusion methods.

The block-based methods usually divide the image into small, regular, and overlapped blocks, and extract robust features from each image block. Finally, by sorting and matching the features, the tampered regions are obtained by marking the image block. Most of the state-of-the-art block-based detection methods exploit the six principal categories based on the block feature extraction technique [1]: 1) Frequency domain-based methods [2]–[4]. These methods are invariant to many post-processing operations such as compression, blurring, and noise. Reference [4] is robust against JPEG compression, blur, and noise effects due to DCT and SVD combination. However, frequency-based block features are variants to a geometric transformation. 2) Dimensionality reduction-based methods [5], [6]. Block feature-based methods will extract a large number of local features, which will cause high time complexity. Dimension reduction techniques have been used to reduce the dimension of extracted block features and increase the matching processing speed. 3) Local binary pattern (LBP)-based methods [7]–[9]. LBP is a grey-scale texture operator which is used to describe the spatial structure of the image texture. Reference [17] is invariant to translation, scaling, and illumination due to the combination of Hessian points and center symmetric LBP (CSLBP). However, this method is not robust against rotation and blur degradation because of the lacking of rotation and blur-invariant features. 4) Texture-based methods [10], [11]. A combination of statistical analysis and color texture to segment the region of interest is given in [18]. However, if the image contains a high degree of blur, the detection rate of this method fell significantly. Texture-based methods are not robust against geometric transformation attacks. 5) Moment invariant-based methods [12], [13], [19]. Moment invariant is a set of features that are invariant to a geometric transformation, including blur moment, Hu moment, Zernike moment and so on. In [13], Quaternion exponent moment (QEM) moduli are extracted from each overlapped circular color block. The main limitation of this method is the higher computational complexity, which can be reduced by applying super pixel theory. 6) Miscellaneous methods [14]–[16].

As can be seen from the above, most block-based methods lack robustness to geometric transformation attacks, and have a large number of features, resulting in high time complexity. Therefore, keypoint-based methods are proposed.

Keypoint-based methods make up for the above shortcomings by extracting the keypoints in the high entropy regions and describing local features. Among them, Scale-Invariant Feature Transform (SIFT) [20]–[24], [29] and Speeded Up Robust Features (SURF) [25]–[27], [30] are widely used in the detection stage for feature extraction.

SIFT was introduced by Lowe [20] and gradually developed in the field of image forgery detection. Mirror-Reflection Invariant Feature Transform (MIFT) is introduced in [21], which has the properties of SIFT features and robust against mirror reflection. In [22], principal component analysis (PCA)-SIFT along with k-nearest neighbors (k-NN) is used. The dimensions of extracted SIFT features are reduced by PCA. The combination of DyWT and SIFT is given in [23]. It is observed that due to the shifting invariant property, DyWT is more accurate than DWT in CMFD and combination of DyWT with SIFT gives more robust results than the

conventional CMFD techniques. Fusion of block and SIFT key point methods is proposed in [24]. In this, the method the image is divided into non-overlapped regions using Simple Linear Iterative Clustering (SLIC) and SIFT keypoints are extracted from all regions. Based on the SIFT keypoints, regions are segmented into smooth and non-smooth regions. Therefore, Zernike moment is used in smooth regions and SIFT is used in non-smooth regions. It is observed that reliability and efficiency of the method are depend on SIFT and Zernike moment, respectively.

SURF is an alternative to the SIFT descriptor, which has a faster matching speed than SIFT due to Hessian matrix's approximation and integral image. In [25], SURF descriptors are extracted from the forged image and matching is performed between the subsets of the descriptor. It is observed that the method is fast as well as reliable in small-sized images. However, localization of forgery is not done. In [26], authors combined SURF and SIFT, which make the forgery detection algorithm fast and robust. SURF is used for fast detection and SIFT is used for robustness of CMFD. For detection of forgery in small smooth regions, a method is given in [27]. In this, the image is segmented into non-overlapped and irregular super pixels. Stable image keypoints are extracted from each super pixel and exponent moment magnitudes, best bin first, and reversed-g2NN algorithm are used for matching. However, it is not suitable for real-time applications since it has higher computational cost.

Based on the advantages of the above two methods, relevant scholars have proposed the fusion methods which combine block-based methods and keypoint-based methods [31], [32]. Ardizzone *et al.* [31] extracted Delaunay triangle features of keypoints, and detected image forgery by matching triangles. Pun *et al.* [32] proposed a CMFD algorithm based on adaptive over-segmentation. They extracted keypoints from each segmentation blocks as block features, and the block features are matched with one another to locate the labeled keypoints. These methods perform well in smooth regions detection and forgery localization. Although some of them adopt super-pixel method, reducing computational complexity is still not their focus.

In recent years, with the development of deep learning, more and more applications using neural networks have greatly improved the computational efficiency and accuracy [41]–[43]. The research of CMFD methods based on neural network has been rising gradually [39], [40]. Liu *et al.* [39] proposed a CMFD method based on Convolutional Kernel Network. It is reformulated as a series of matrix computations and convolutional operations which are easy to parallelize and accelerate by GPU, leading to high efficiency. Wu *et al.* [40] introduced BusterNet which is an end-to-end trainable, deep neural network solution. This is the first CMFD algorithm with discernibility to localize source/target regions. Although the method based on neural network is suitable for image processing in a novel way, it needs more samples for learning and training, and its accuracy is slightly lower than those traditional methods which focus on the pixel level detection rate of image CMFD. Therefore, this paper is compared with the traditional method.

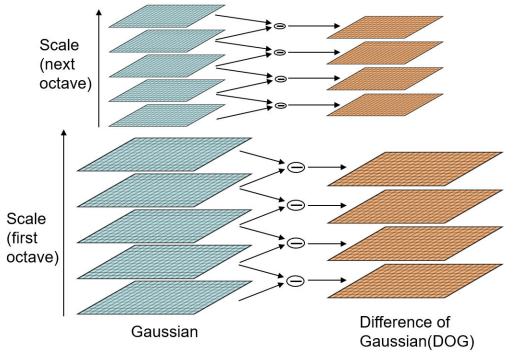

FIGURE 2. Structure of the Gaussian scale space images (left) and DoG images (right).

III. SIFT FEATURE REPRESENTATION

SIFT is a classic and still popular feature descriptor in the field of forgery detection. The essence of SIFT algorithm is to find keypoints (feature points) in different scale spaces and calculate the dominant orientation of keypoints. The key points found by SIFT are some very prominent points, such as corners, edges, bright spots in dark areas, and dark spots in bright areas, which are not changed by illumination, affine transformation, and noise. The main computational stages for generating image feature sets are as follows [20]:

A. SCALE-SPACE EXTREMA DETECTION

The first stage of computation searches over all scales and image locations. As shown in Fig.2, it is implemented efficiently by using a Difference-of-Gaussian (DoG) function to identify potential interest points that are invariant to scale and orientation. Successive Gaussian-blurred images $L(x, y, \sigma)$ are generated by repeatedly convolving the given image $I(x, y)$ with Gaussian filters at multiple scales, $G(x, y, \sigma)$ as follows:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{1}$$

where $*$ is the convolution operation in x and y, and

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{2}$$

Then, the SIFT keypoints are selected as local extrema within a $3 \times 3 \times 3$ cube of the DoG domain. Specifically, the DoG image at scale σ is given by

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \tag{3}$$

B. KEYPOINT LOCALIZATION

The candidate keypoints detected in the previous stage are extremum points in discrete space. The position and scale of the keypoints can be determined accurately by fitting the three-dimensional quadratic function. Meanwhile, due to the DoG operator produces strong edge response, the low

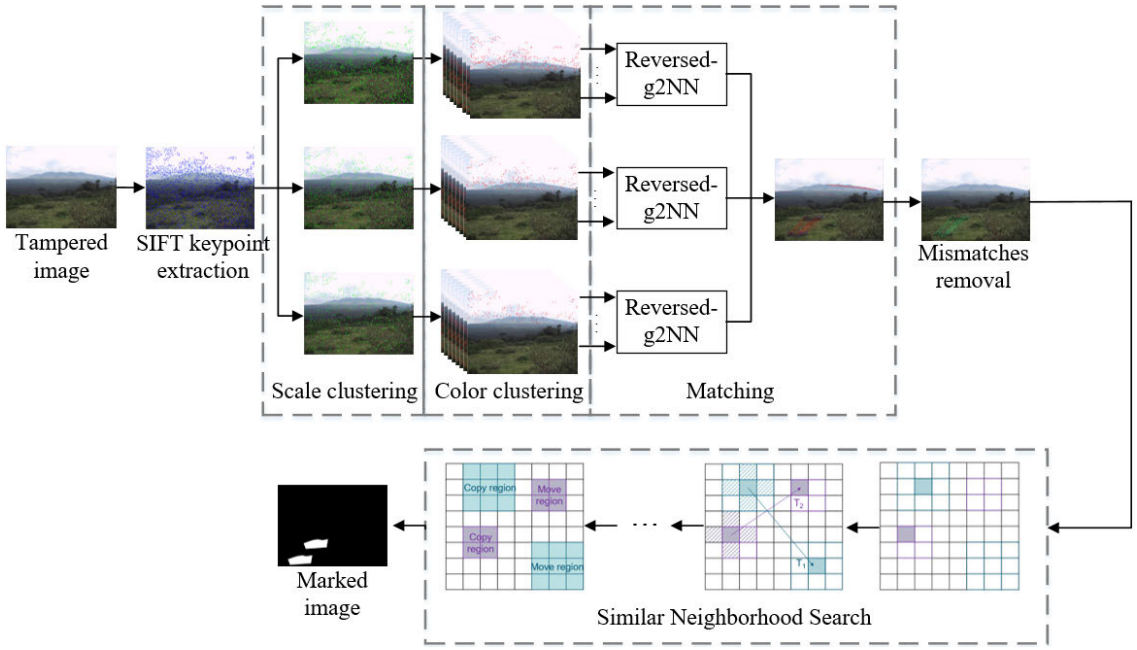

FIGURE 3. The framework of the proposed method; The specific process includes feature extraction, keypoints clustering, feature matching, mismatches removal and tampered region location.

contrast keypoints and unstable edge response points can be rejected in order to enhance matching stability and anti-noise ability.

C. ORIENTATION ASSIGNMENT

In order to make the descriptor rotation invariant, it is necessary to assign a dominant orientation to each keypoint by using the local features of the image. The stable dominant orientation of local structure is obtained by calculating image gradient. For each point (x, y, σ), its orientation is computed as

$$\begin{aligned} \theta(x, y) &= \tan^{-1}\left(\frac{dy}{dx}\right) \\ dy &= L(x, y+1) - L(x, y-1) \\ dx &= L(x+1, y) - L(x-1, y) \end{aligned} \tag{4}$$

where dy and dx are the vertical and horizontal gradients of keypoint. Then, the histogram is used to calculate the gradient and orientation of the pixels in the neighborhood. The peak value of histogram represents the orientation of the neighborhood gradient at the keypoint, and the maximum value of the histogram is the dominant orientation of the keypoint.

D. KEYPOINT DESCRIPTOR

It is suggested that the gradient information of eight directions calculated in the neighborhood of 4×4 in the keypoint scale space is used to represent the SIFT descriptor with a total of $4 \times 4 \times 8 = 128$ dimension vectors.

Through the above four phases, a list of n keypoints $\{k_1, k_2, \ldots, k_n\}$ and their corresponding descriptors $\{f_1, f_2, \ldots, f_n\}$ are generated for a given image $I(x, y)$.Let k be a SIFT keypoint descriptor, which is represented as a quaternion vector:

$$k = (x_k, y_k, \sigma_k, \theta_k) \tag{5}$$

where (x_k, y_k) are the coordinates in the image plane, σ_k denotes the scale and θ_k serves as its dominant orientation.

IV. PROPOSED METHOD

Following the traditional CMFD process, an efficient method is designed based on SIFT keypoints scale-color clustering and similar neighborhoods searching. The overall framework of the proposed method is shown in Fig.3.

The proposed method consists of feature extraction, keypoints clustering, feature matching, mismatches removal and tampered region location. Firstly, SIFT was selected to describe keypoints. Then we cluster the keypoints based on scale and color to put them into several smaller clusters which have been matched separately. J-Linkage algorithm is used to remove mismatches and estimate affine transformation matrix. Finally, similar neighborhoods of matching pairs are searched to locate tampered regions.

A. KEYPOINTS SCALE-COLOR CLUSTERING

Due to the large number of SIFT keypoints, especially in high resolution images, there are two problems in matching: 1) The time complexity of feature matching is $O(n^2)$, where n is the number of keypoints. When the value of n is larger, the time complexity will increase in square order; 2) In the fact, correct matching pairs account for a small proportion. In matching, each point needs to be compared with other n-1 points, but most of them are unnecessary. Therefore, it is necessary to cluster keypoints before feature matching in order to reduce the computational complexity of matching.

In order to solve the above problems, this paper improves the hierarchical feature point matching algorithm in [29]. The clustering framework based on SIFT keypoints scale-color is

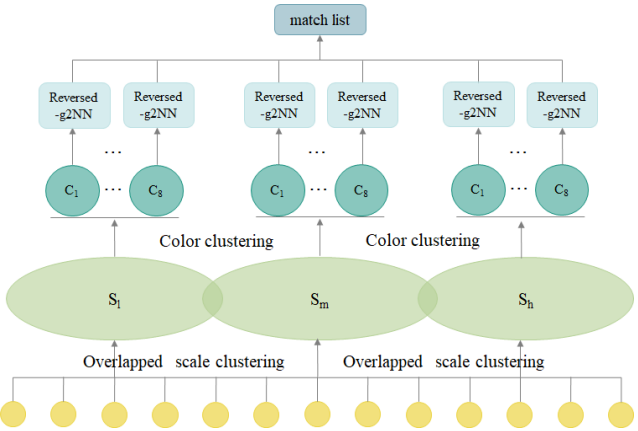

FIGURE 4. Clustering based on scale and color of keypoint.

shown in Fig. 4. It consists of two parts: 1) clustering based on overlapped scale; 2) clustering based on color of keypoints.

1) CLUSTERING BASED ON OVERLAPPED SCALE

SIFT keypoints are extreme points in Gaussian scale space, as shown in Fig.2. Each octave of keypoints has its corresponding scale, that is, the third element σ_k in the descriptor. Most of the keypoints that can be matched have similar scales, that is, there are few cases where the scales of the two keypoints in matching pairs are quite different. Therefore, the main purpose of this step is to maximize the separation of keypoints of different scales in order to reduce unnecessary matching and comparison. We know that the number of large-scale keypoints is much smaller than that of small-scale keypoints, so we group up the small-scale range and retain the large-scale range. At the same time, scale clustering is performed via overlapped scale to maintain the robustness of scaling attack tampering.

Specifically, we can get the scale σ_k of keypoint in the feature extraction stage. According to the value of scale σ_k, the keypoints are divided into three groups, which are expressed by S_l, S_m, and S_h respectively. The strategy of grouping is to divide all scales into three groups: low, medium and high. High scales are separated into a group, and we divide the small scales below 4 into low and medium groups respectively. At the same time, there is a 0.5 range coincidence between each group to ensure scale invariance in a certain range.

$$\begin{aligned} S_l &= \{k_i \,|\, 0 \le \sigma_{k_i} < 3, i = 1, \ldots, n\} \\ S_m &= \{k_i \,|\, 2.5 \le \sigma_{k_i} < 4, i = 1, \ldots, n\} \\ S_h &= \{k_i \,|\, \sigma_{k_i} \ge 3.5, i = 1, \ldots, n\} \end{aligned} \tag{6}$$

We divide the small scales below 4 to meet $n_l^2 + n_m^2 + n_h^2 < n^2$, and the matching time complexity has been significantly reduced. However, the disadvantage of scale clustering is that it sacrifices the robustness of some scaling attacks, because matching can only be done within the group, and cannot be matched between S_l, S_m, and S_h. For this reason, we overlap the boundaries in a certain range, and the larger scale keypoints are concentrated in the S_h, which guarantees the robustness of scaling attack to a certain extent.

	[0,127]	[128,255]
R	R1	R2
G	G1	G2
B	B1	B2

FIGURE 5. RGB channel partition.

2) CLUSTERING BASED ON COLOR

After scale clustering, a large number of low-scale keypoints will be clustered in S_l and S_m. Color-based clustering is performed in order to further reduce the time complexity. Although image has been attacked by geometric transformation or post-processing operation, the corresponding matching points will not have much difference in color due to the similarity between the tampered region and the source region. If color image is converted to gray image, different RGB values may be converted to the same gray value, which may cause some degree of inaccuracy and redundancy. However, the RGB value of the keypoint in the tampered region does not change much, which can preserve the visual features of the image as much as possible. Therefore, based on the previous step, we further adopt the color-based clustering method, that is, color clustering was carried out in S_l, S_m, and S_h groups respectively.

Three color channels of RGB are divided into two parts, as shown in Fig. 5. Eight combinations are obtained by arranging the six parts as shown in (7). All the keypoints in each scale group are divided into eight small groups with a total of $3 \times 8 = 24$ groups. The keypoints in twenty-four groups are matched respectively.

$$\begin{aligned} C_1 &= \{k_i \,|\, r_{k_i} \in R_1 \wedge g_{k_i} \in G_1 \wedge b_{k_i} \in B_1, i = 1, \ldots, n\} \\ C_2 &= \{k_i \,|\, r_{k_i} \in R_1 \wedge g_{k_i} \in G_2 \wedge b_{k_i} \in B_1, i = 1, \ldots, n\} \\ &\cdots \\ C_7 &= \{k_i \,|\, r_{k_i} \in R_2 \wedge g_{k_i} \in G_2 \wedge b_{k_i} \in B_1, i = 1, \ldots, n\} \\ C_8 &= \{k_i \,|\, r_{k_i} \in R_2 \wedge g_{k_i} \in G_2 \wedge b_{k_i} \in B_2, i = 1, \ldots, n\} \end{aligned} \tag{7}$$

where r_{k_i}, g_{k_i}, and b_{k_i} represent R, G, and B channel values corresponding to keypoints, respectively.

Compared with reference [29], this method can effectively reduce the impact of scale changes on detection results by clustering keypoints on overlapping scales. And color based clustering is more accurate than gray. We experimented on ten high-resolution images and ten low-resolution images in the data set, and counted the matching time without clustering and with clustering, respectively. The statistical results are shown in Table 1. From Table 1, it can be seen that the matching time after clustering is significantly shortened and the time complexity is effectively reduced.

TABLE 1. Matching time comparison.

	without clustering	with clustering
High resolution images	62.15s	**34.56s**
Low resolution images	0.94s	**0.29s**

FIGURE 6. Starting from the first column: 1: forged images, 2: clustering results with HAC.

B. REVERSED-G2NN FEATURE MATCHING

Traditional g2NN [28] can detect multiple copy-move forgery effectively, calculate the distance set $D = \{d_1, d_2, \ldots, d_{n-1}\}$ between keypoint k_i and other (n-1) keypoints in the cluster, and rank them in ascending order. When (8) is satisfied, k_i is matched with the keypoints corresponding to d_1.

$$\frac{d_1}{d_2} \leq t, \quad where\ t \in (0, 1) \tag{8}$$

However, g2NN may omit some correct matching pairs when the multiple copy-move forgery regions are very similar. Based on this, Reversed-g2NN [34] is used for feature matching.

Similar to g2NN, the distance set of keypoints is calculated first, and the distance ratio $T_i = d_{i-1}/d_i, i = 10$ is calculated in reverse order. If T_j is larger than the threshold t and T_{j-1} is smaller than t, then k_i matches with the keypoints set $\{k_1, k_2, \ldots, k_j\}$ corresponding to $\{d_1, d_2, \ldots, d_j\}$. In order to minimize mismatches, threshold t in this paper is taken as 0.45 experimentally.

C. MISMATCHING REMOVAL AND ESTIMATING AFFINE TRANSFORMATION

The clustering-based method, like HAC, for removing mismatches is performed by taking into account only the coordinates of the matched pairs and not the matching constraint between points. There are two main drawbacks in this kind of clustering on spatial location method [35]: 1) the inability to separate duplicated regions that are close to each other and 2) the difficulty to identify a patch as single, when it contains keypoints with a non-uniform spatial distribution. Take HAC for example, as shown in Fig. 6. Based on the above problems, we apply the robust clustering algorithm J-Linkage [36] to remove mismatches and estimate affine transformations.

D. TAMPERED REGIONS LOCALIZATION

The matching pair based on keypoints cannot locate the tampered regions accurately, and it is difficult to quantify the experimental data. The tampering regions which have been segmented by localization algorithm can be visually distinguished, and the experimental results can be quantified by comparing with the truth image.

Based on this, we propose a localization algorithm via searching similar neighborhoods, which takes pixels as the smallest unit and compares them one by one, similar to the region growing algorithm. The main ideas of the algorithm are as follows: Adding the first keypoint to the expansion queue, and the head element of the queue is set as the seed point, which is affine transformed to find its corresponding point. Comparing PCT feature and PSNR of seed point and corresponding point, if both are satisfied, they are marked, and the eight neighborhoods of seed point are added to the expansion queue and deleting seed point at the same time. Otherwise, the seed point will be deleted directly and the head element of the queue will be selected as the seed point to continue to expand. The pseudocode of the algorithm is shown in Algorithm 1.

Algorithm 1 Similar Neighborhood Search and Localization Algorithm

Input: Matching list **ML,** Affine Transformation list **T**
Output: Marked localization image **Result**

```
while ( T != null )
  Create an empty expansion queue Q;
  Select the first matrix T1 in T;
  Select the first matching list ML1 in ML;
  while ( ML1 != null)
    ka = left point of the first matching point pair;
    if ( ka is unmarked)
      Add ka in Q;
      Delete corresponding point pairs in ML1;
    else
      Delete corresponding point pairs in ML1;
      continue;
    end if
    while ( Q != null )
        kb = head element in Q;
        k'b = T1 * kb;
        if ( | PCTkb - PCTkb' | < ta && | PSNRkb - PSNR
kb' | > tb )
        Mark kb and k'b;
        Add eight neighborhoods of kb to Q;
        Delete kb in Q;
      end if
    end while
  end while
  Delete T1;
  Delete ML1;
end while
```

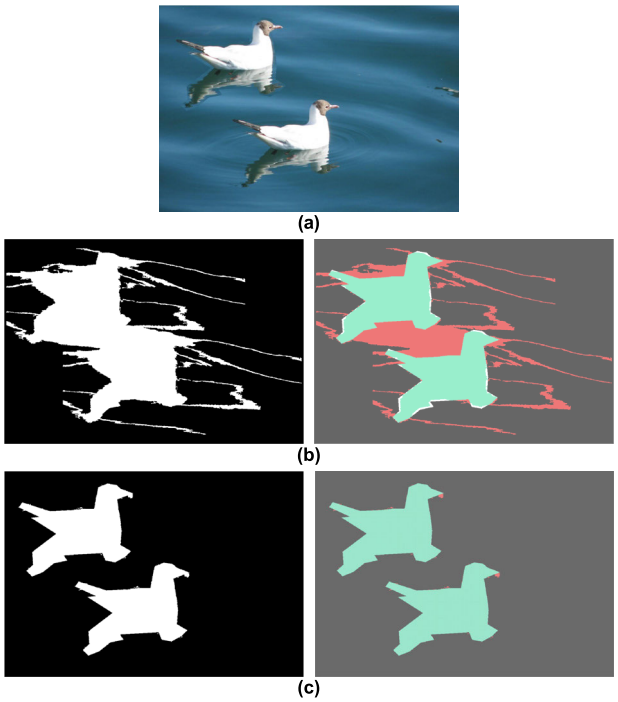

FIGURE 7. Comparative results: (a) tampered image; (b) experimental results with PCT; (c) experimental results with PCT and PSNR.

The PCT feature transformation core is orthogonal, so the feature extracted by PCT is more compact than that calculated by non-orthogonal kernel, and the PCT feature extraction is to extract each feature from the circular convolution core with diameter B. The convolution core slides one pixel along the image from top left to bottom right for feature extraction, which is computationally efficient and robust to rotation attacks. However, only using PCT features will cause problems as shown in Fig. 7. (c). Therefore, we introduce PSNR and PCT features together as similarity metric for searching and localization. As shown in Fig. 7, (a) for forgery image, (b) for experimental results using PCT as similarity metric only, binary image for experimental results, red-green image for the result of comparison with the ground truth image, where green is the correctly detected tampered region, red denotes the false detected region, white represents the missed detection region, and grey represents the real untampered region. (c) show the results of PCT and PSNR as similarity measure. From this result, we can see that the experimental results which using PCT and PSNR as similarity measure are much better than those using PCT only.

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. DATASET AND EXPERIMENTAL ENVIRONMENT

The experiments are conducted on the tampered images of three public domain benchmark databases: GRIP [16], Dataset [31], and FAU [37] which all consist of tampered images and corresponding ground truth images. The GRIP database contains $2 \times 80 = 160$ ground truth images and tampered images which tampered regions have arbitrary shape, ranging in size from 4000 pixels (less than 1% of the image) to 50000 pixels. Dataset consists of three parts: D0:

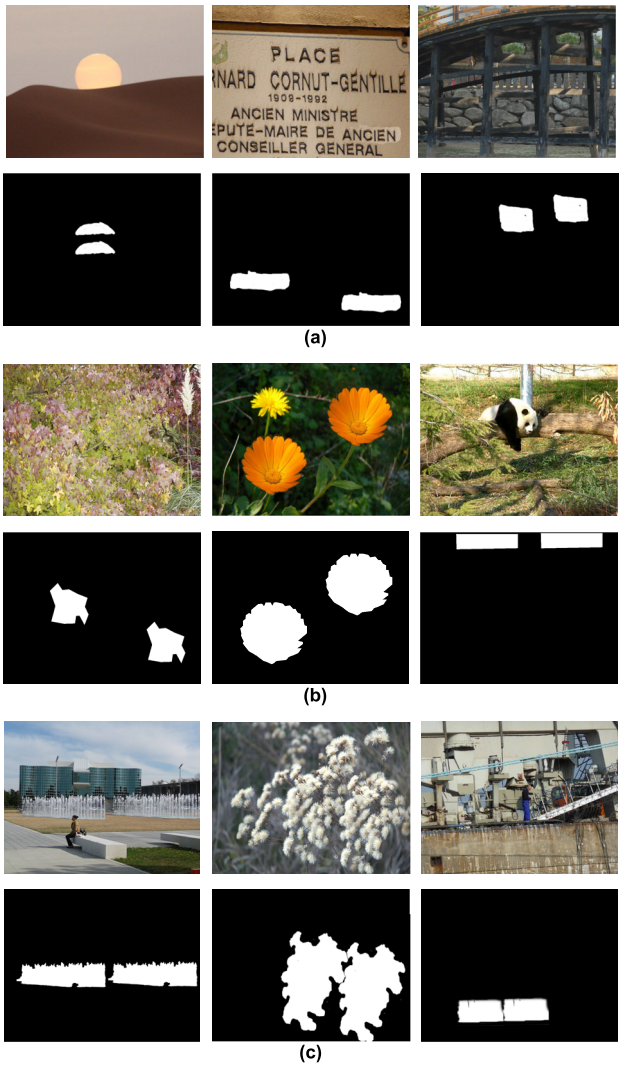

FIGURE 8. Examples for three datasets: (a) GRIP; (b) Dataset; (c) FAU.

fifty tampered images only after simple translation; D1 and D2: twenty groups of images after a series of rotation and scaling; D3:D0 corresponding to fifty original images without tampering. FAU is a classical dataset which contains forty-eight basic images. The average size of tampered region is about 10% of each image. Tampering means include rotation, scaling, JPEG compression and noise. The images in the first two datasets are all small in resolution, whose size is about 1000×700, and the image resolution in the third dataset is large, and the image size is about 3000×2000. Fig. 8 shows the tampered images and corresponding ground truth images in three datasets, in which white is the tampered region and black is the real region.

All the experiments are conducted on a machine with Intel(R) Core (TM) i5-6200U CPU @ 2.30GHz 2.40GHz, 8GB RAM and runs on Matlab R2018a.

B. EVALUATION METRICS

The performance of the proposed method is measured at both image level and pixel level. At the image level,

TABLE 2. Time comparison of the proposed method.

Datasets	FE(s)	FM(s)	FC(s)	FL(s)	Total(s)			
					Proposed	Hierarchical [29]	Iterative [38]	PM-ZM-Polar [16]
Dataset	1.19	0.62	3.60	7.02	**12.44**	33	71.6	11.7
GRIP	1.25	0.95	4.02	4.10	**9.76**	13.9	25.7	12.05
FAU	7.62	85.29	288.71	80.97	**464.80**	86.6	468.2	244.7

TABLE 3. Experimental results of simple forgery in dataset.

Methods	Image level			Pixel level		
	TPR(%)	FPR(%)	F_1-image(%)	Precision(%)	Recall(%)	F_1-pixel(%)
Hierarchical [29]	98.00	2.00	98.00	88.36	96.09	91.45
Iterative [38]	100	36.00	84.75	73.52	99.05	81.40
PM-ZM-Polar [16]	96.00	2.00	96.97	89.02	98.48	93.33
Proposed	**90.00**	**2.00**	**93.75**	**99.58**	**97.11**	**98.07**

TABLE 4. Experimental results of simple forgery in GRIP.

Methods	Image level			Pixel level		
	TPR(%)	FPR(%)	F_1-image(%)	Precision(%)	Recall(%)	F_1-pixel(%)
Hierarchical [29]	100	0	100	-	-	94.66
Iterative [38]	100	33.75	85.56	-	-	66.44
PM-ZM-Polar [16]	98.75	6.25	97.53	95.90	96.41	96.15
Proposed	**90.00**	**10.42**	**91.72**	**99.79**	**99.67**	**99.72**

TABLE 5. Experimental results of simple forgery in FAU.

Methods	Image level			Pixel level		
	TPR(%)	FPR(%)	F_1-image(%)	Precision(%)	Recall(%)	F_1-pixel(%)
Hierarchical [29]	100	2.08	98.97	-	-	94.28
Iterative [38]	100	52.08	79.34	-	-	86.07
PM-ZM-Polar [16]	-	-	94.95	-	-	93.72
Proposed	**100**	**0**	**100**	**99.96**	**98.59**	**99.24**

we focus on the ability of images to be correctly classified as forged or authentic, and only detect on simple copy-move forgery. At the pixel level, we analyze the performance of locating tampered regions accurately to verify the robustness of the method. In this paper, by regarding the tampered images/pixels as positive samples and the authentic images/pixels as negative ones, the *True Positive Rate* (*TPR*), *False Positive Rate* (*FPR*), and F_1 are utilized to evaluate the performance of the proposed methods. *TPR* represents the proportion of actual tampered images in the detection results and it can also be called Recall rate. We expect the higher the value, the better. *FPR* represents the proportion of the number of real images that have been mistakenly detected as tampering. We expect it to be as low as possible. F_1 is a comprehensive evaluation index, which is regarded as a harmonic average of precision and recall rate. The higher value of F_1, the better experimental results can be reflected. *TPR*, *FPR*, and F_1-*image* are used at the image level, and F_1-*pixel* is used at the pixel level. They are defined as follows:

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

$$F_1 = \frac{2TP}{2TP + FP + FN} \tag{9}$$

where *TP* (True Positive) is the number of tampered images/tampered pixels classified as tampered (correct detection); *FP* (False Positive) is the number of authentic images/authentic pixels classified as tampered (false detection); *TN* (True Negative) is the number of authentic images/authentic pixels classified as authentic (correct detection); and *FN* (False Negative) is the number of tampered images/tampered pixels classified as authentic (omitted detection).

C. METHOD FOR COMPARISON

Our experimental results of the proposed method are compared with three recently mainstream detection algorithms that are: Hierarchical matching (2019) [29], Iterative strategy (2016) [38], and Patch Match (2015) [16].

Reference [29] proposed a hierarchical feature point matching method based on SIFT keypoint. Reference [38] proposed an iterative improvement strategy based on the new interest point detector. The whole procedure is iterated along

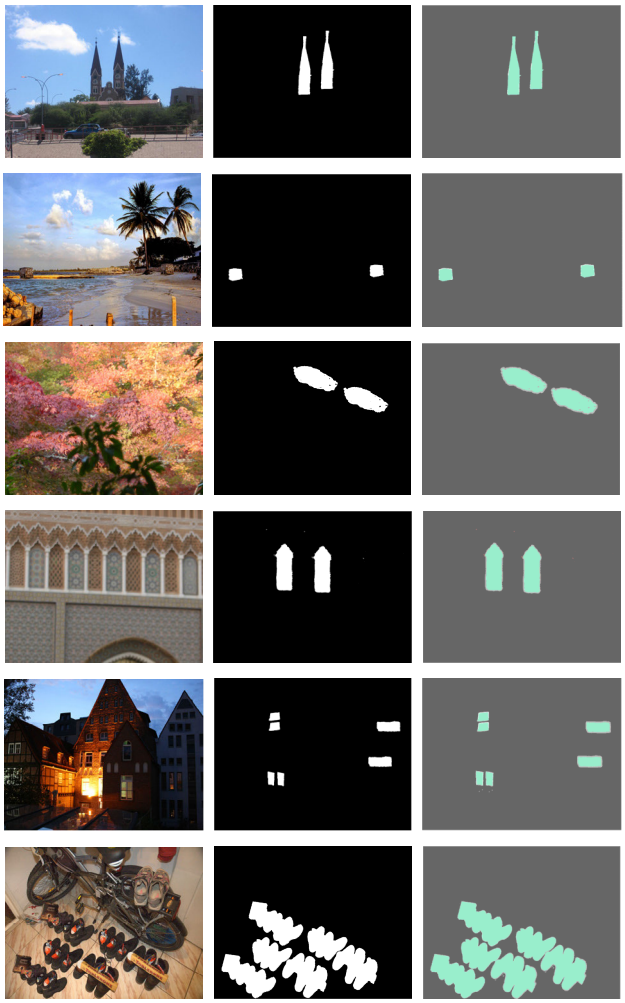

FIGURE 9. Experimental results of simple forgery. Starting from the first column: 1: forged images used in experiments, 2: experimental results, 3: comparisons with ground truth images.

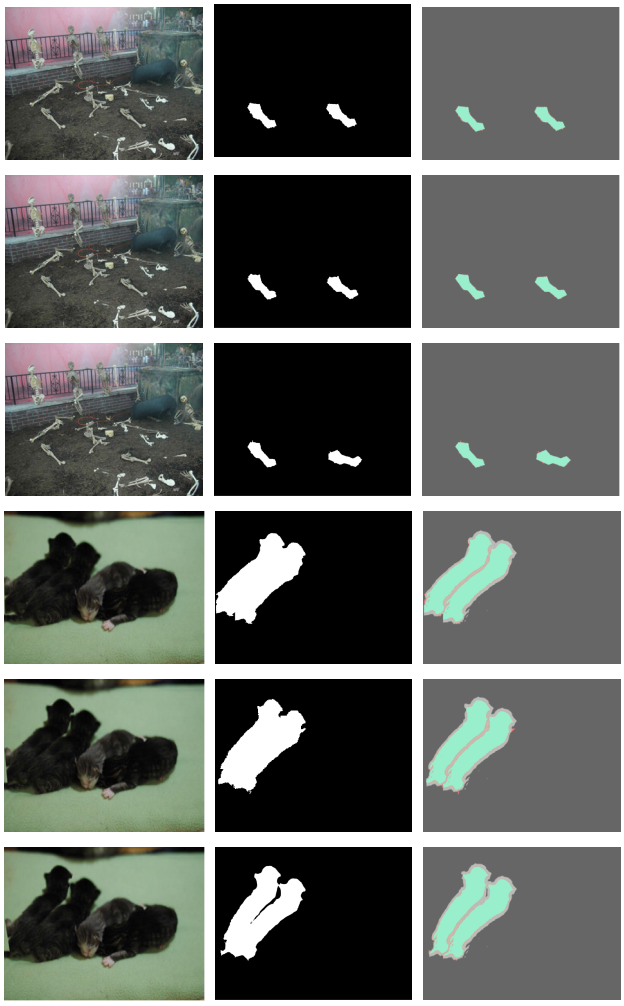

FIGURE 10. Experimental results of rotation transformation forgery. Rotational angles are 5°, 10°, 30° and 2°, 6°, 10°, respectively.

with adjusting the keypoints density based on the achieved information. Reference [16] is an efficient algorithm based on rotation-invariant features computed densely on the image. It applied a fast approximate nearest-neighbor search algorithm, Patch Match, especially suited for the computation of dense fields over images. We use Zernike Moment in polar coordinates as a feature. All the source codes for the comparison experiment have been published online. The parameters in the comparison methods are set to the default values.

D. EXPERIMENTAL RESULTS AND COMPARATIVE ANALYSIS

In this section, experiments are carried out to detect the effectiveness of the proposed method, showing the detection results after geometric and post-processing tampering. Among all the experimental results, the first column is tampered image, the second is the binary representation of the experimental results, and the third is the comparison between the experimental results and the ground truth image. where green denotes the correctly detected tampered regions, red denotes the wrong detection regions, white denotes the omitted detection regions, dark gray denotes the authentic untampered regions, and light gray denotes the tampered region boundaries which are post-processed after tampering. The validity of the proposed method can be verified by comparing the difference between the experimental results and the ground truth images.

1) SIMPLE FORGERY DETECTION

We have experimented on three datasets, in which the images only undergo simple translation transformation. Table 2 shows the time complexity of each stage and the comparison between this paper and the existing methods. Among them, each stage is: FE (Feature Extraction), FM (Feature Matching), FC (Feature Clustering) and FL (Forgery Localization). It can be seen that compared with other methods, this method has obvious advantages in total time, especially in low resolution image database. However, the method in this paper takes a lot of time in the forgery localization and feature

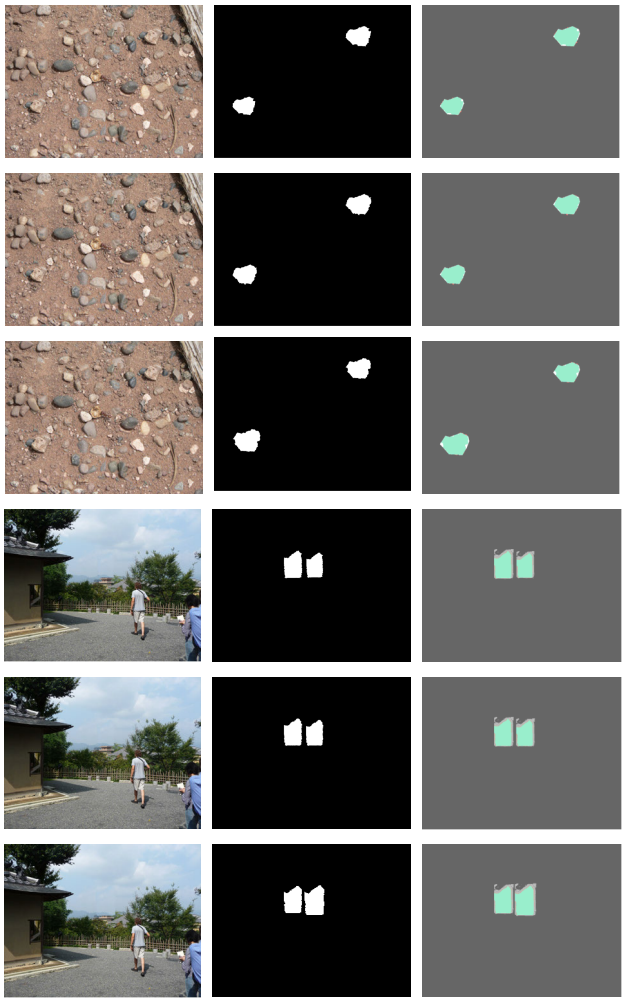

FIGURE 11. Experimental results of scaling transformation forgery. Scaling factors are 0.9, 0.95, 1.1 and 0.91, 0.95, 1.09, respectively.

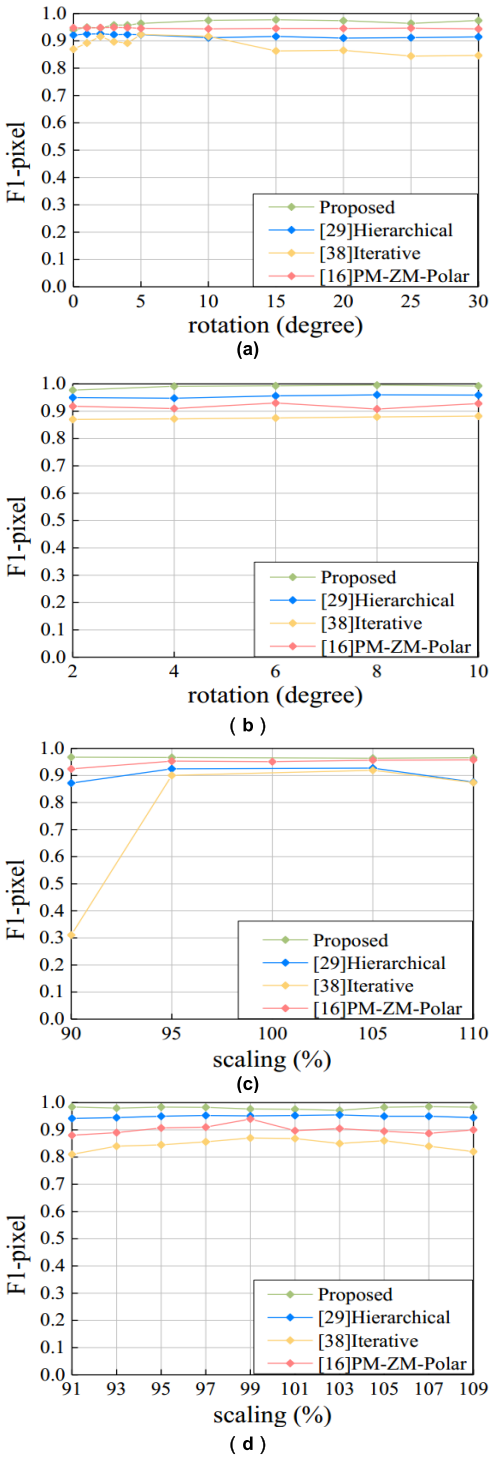

FIGURE 12. The comparative analysis results of geometric transformation forgery detection are as follows: (a) and (b) are the results of rotation forgery detection in Dataset and FAU, (c) and (d) are the results of scaling forgery detection in Dataset and FAU, respectively.

clustering stage, mainly because some tampered regions are large, resulting in a large number of keypoints, and the time complexity is proportional to the size of tampered regions especially in the high resolution image set FAU. Table 3, 4 and 5 give the evaluation results of the method at image level and pixel level, and the experimental data are averaged. It can be seen that the method in this paper performs well on all three datasets. At the image level, in the two small resolution datasets, the method has a certain degree of omitted detection, but the false detection rate is relatively low, so the F_1-score at the image level is slightly lower than [29] and [16], higher than [38]. In the large resolution dataset FAU, the method performs better, and each metric is better than other three methods. At the pixel level, the accuracy is over 99%, and the F_1-score is 98.07% in the Dataset, and the others are above 99%, which is significantly improved compared with the other three methods. The reason why the proposed method and literature [29] perform well is that they can effectively remove the mismatch, and the forgery localization algorithm of the proposed method performs on the pixel level iteratively, which is fine and accurate. Reference [16] extracts rotation-invariant features computed densely on the image, so the detection accuracy is guaranteed. However, the literature [38] selects a large number of interest points. After a finite number

FIGURE 13. Experimental results of JPEG compression forgery. Compression factors are 90, 70 and 50.

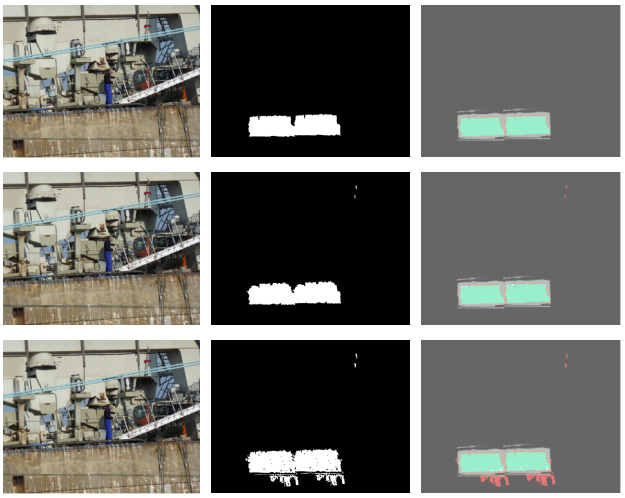

FIGURE 14. Experimental results of noise forgery. The standard deviations of noise are 0.02, 0.06 and 0.1.

of iterations, as many points of interest are identified as tampering regions as much as possible, so there will be more mismatches, resulting in higher TPR and recall rate, but lower precision rate. Some experimental results are shown in Fig. 9.

2) GEOMETRIC TRANSFORMATION FORGERY DETECTION

Geometric forgery means include rotation transformation forgery and scaling transformation forgery. As the large-scale scaling and rotation experiments fail to achieve the expected goal, we will continue to study in the future. This paper only verifies the forgery of small-scale geometric transformation.

a: ROTATION TRANSFORMATION FORGERY DETECTION

This paper experiments on the images of the tampered region that have undergone rotation transformation in the Dataset and FAU. The rotation angles of Dataset tampered region are $\pm$ 1° to $\pm$ 5°, the step length is 1°, and 10° to 30°, the step length is 5°. The rotation angles of FAU tampered region are 2° to 10°, and the step length is 2°. The experimental results are shown in Fig. 10.

b: SCALING TRANSFORMATION FORGERY DETECTION

This paper experiments on the images of the tampered region that have undergone scaling transformation in the Dataset and FAU. The scaling size of tampered region in Dataset ranges from 0.2 times to 2 times, in FAU ranges from 0.91 times to 1.09 times, step size is 0.02 times. Fig. 11 shows the experimental results.

The experimental comparison and analysis are shown in Fig. 12. In rotation transformation forgery detection, we can see that the performance of the four methods are relatively stable, especially for small-scale rotation, the F_1-scores of our algorithm is more than 95%, approaching 1, [16] performs well, [38] performs relatively weakly. In the scaling transformation forgery detection, the four methods have good performance, but the performance of literature

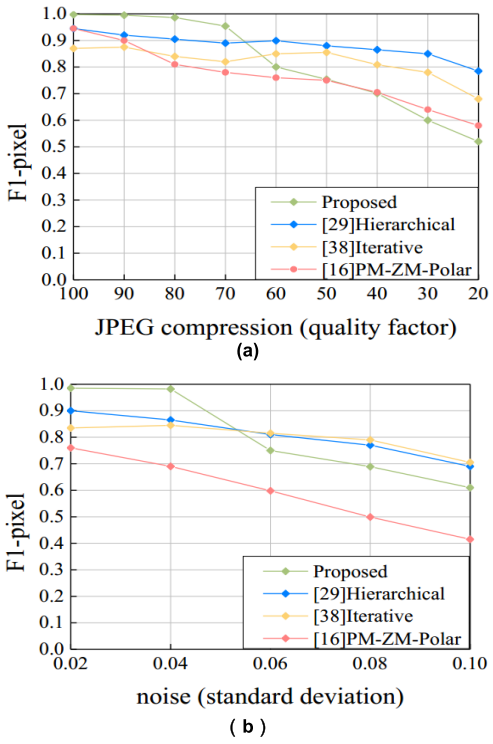

FIGURE 15. The comparative analysis results of post-processing forgery detection are as follows: (a) and (b) are the results of JPEG compression and noise forgery detection in FAU, respectively.

[38] is poor. When the scaling degree is less than 90%, some images cannot be correctly classified. The other three methods are relatively stable. The proposed method has a good performance in geometric transformation forgery detection due to the invariance of SIFT feature.

3) POST-PROCESSING FORGERY DETECTION

Post-processing forgery means include JPEG compression forgery and noise forgery. We will detect these two means.

a: JPEG COMPRESSION FORGERY DETECTION

In order to verify the detection performance of the proposed method for JPEG compression forgery images, we experimented with tampered images in FAU. The JPEG compression factor is 20 to 100 and the step size is 10. The experimental results are shown in Fig. 13.

b: NOISE FORGERY DETECTION

We have experimented with tampered images in FAU with normalized intensity between 0 and 1, and added zero mean Gaussian noise. The standard deviation varies from 0.02 to 0.10, and the step size is 0.02. Fig. 14 shows the experimental results.

The comparative analysis of the experiment is shown in Fig. 15. The detection results are good in the case of less tampering, but the proposed method performs poorly when the quality factor is lower than 70 and the standard deviation is higher than 0.04. The main reason is that the PSNR is used as one of the similarity measures in the localization algorithm, which is sensitive to the change of post-processing tampering factor, resulting in poor experimental result. The other three methods are relatively stable, although their performance are not as good as those of the proposed methods. How to improve the detection of post-processing forgery is also a problem we need to solve in the future.

VI. CONCLUSION

In this paper, we present a novel CMFD method based on SIFT keypoint to locate the doctored regions at the pixel level, and the experimental results show that the method has performed well. SIFT keypoint has a high dimension and usually extracts tens of thousands or even hundreds of thousands keypoints for an image, which impose a huge burden on feature matching. By clustering the SIFT keypoints, this method can effectively reduce the matching time and improve the detection efficiency. In addition, aiming at the problem that the detection results of keypoint method cannot locate the tampered regions accurately, we propose an algorithm of searching similar neighborhood to iteratively mark the image at pixel level and produce the final tampered localization map. The proposed method is evaluated on three manipulated datasets. Compared with the existing methods, our method has better robustness and could accurately locate tampered regions, especially on simple forgery, geometric transformation forgery and small-scale post-processing forgery. However, in large-scale forgery, this method is not stable and has poor effect, which is also the problem we need to solve in the future.

REFERENCES

[1] B. Soni, P. K. Das, and D. M. Thounaojam, "CMFD: A detailed review of block based and key feature based techniques in image copy-move forgery detection," *IET Image Process.*, vol. 12, no. 2, pp. 167–178, Feb. 2018.
[2] S. A. Fattah, M. M. I. Ullah, M. Ahmed, I. Ahmmed, and C. Shahnaz, "A scheme for copy-move forgery detection in digital images based on 2D-DWT," in *Proc. IEEE 57th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, College Station, TX, USA, Aug. 2014, pp. 801–804.
[3] G. Muhammad, M. Hussain, and G. Bebis, "Passive copy move image forgery detection using undecimated dyadic wavelet transform," *Digit. Invest.*, vol. 9, no. 1, pp. 49–57, Jun. 2012.
[4] 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.
[5] T. Zhang and R.-D. Wang, "Copy-move forgery detection based on SVD in digital image," in *Proc. 2nd Int. Congr. Image Signal Process.*, Tianjin, China, Oct. 2009, pp. 1–5.
[6] M. Bashar, K. Noda, N. Ohnishi, and K. Mori, "Exploring duplicated regions in natural images," *IEEE Trans. Image Process.*, to be published.
[7] G. Muhammad, M. H. Al-Hammadi, M. Hussain, A. M. Mirza, and G. Bebis, "Copy move image forgery detection method using steerable pyramid transform and texture descriptor," in *Proc. Eurocon*, Zagreb, Croatia, Jul. 2013, pp. 1586–1592.
[8] G. Muhammad, M. H. Al-Hammadi, M. Hussain, and G. Bebis, "Image forgery detection using steerable pyramid transform and local binary pattern," *Mach. Vis. Appl.*, vol. 25, no. 4, pp. 985–995, Sep. 2013.
[9] A. Alahmadi, M. Hussain, H. Aboalsamh, G. Muhammad, G. Bebis, and H. Mathkour, "Passive detection of image forgery using DCT and local binary pattern," *Signal, Image Video Process.*, vol. 11, no. 1, pp. 81–88, Apr. 2016.
[10] E. Ardizzone, A. Bruno, and G. Mazzola, "Copy-move forgery detection via texture description," in *Proc. 2nd ACM Workshop Multimedia Forensics, Secur. Intell. (MiFor)*, Firenze, Italy, Oct. 2010, pp. 58–64.
[11] M. Alhussein, "Image tampering detection based on local texture descriptor and extreme learning machine," in *Proc. UKSim-AMSS 18th Int. Conf. Comput. Model. Simul. (UKSim)*, Cambridge, U.K., Apr. 2016, pp. 196–199.
[12] S. J. Ryu, M. J. Lee, and H. K. Lee, "Detection of copy-rotate-move forgery using Zernike moments," in *Proc. Int. workshop Inf. Hiding*, Berlin, Germany, 2010, pp. 51–65.
[13] X.-Y. Wang, Y.-N. Liu, H. Xu, P. Wang, and H.-Y. Yang, "Robust copy–move forgery detection using quaternion exponent moments," *Pattern Anal. Appl.*, vol. 21, no. 2, pp. 451–467, Oct. 2016.
[14] J. Singh and B. Raman, "A high performance copy-move image forgery detection scheme on GPU," in *Proc. Int. Conf. Soft Comput. Problem Solving (SocProS)*, New Delhi, India, Dec. 2011, pp. 20–22.
[15] A. D. Warbhe, R. V. Dharaskar, and V. M. Thakare, "A scaling robust copy-paste tampering detection for digital image forensics," *Procedia Comput. Sci.*, vol. 79, pp. 458–465, Jan. 2016.
[16] D. Cozzolino, G. Poggi, and L. Verdoliva, "Efficient dense-field Copy–Move forgery detection," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 11, pp. 2284–2297, Nov. 2015.
[17] D. M. Uliyan, H. A. Jalab, and A. W. A. Wahab, "Copy move image forgery detection using hessian and center symmetric local binary pattern," in *Proc. IEEE Conf. Open Syst. (ICOS)*, Bandar Melaka, Malaysia, Aug. 2015, pp. 7–11.
[18] D. M. Uliyan, H. A. Jalab, A. W. A. Wahab, P. Shivakumara, and S. Sadeghi, "A novel forged blurred region detection system for image forensic applications," *Expert Syst. Appl.*, vol. 64, pp. 1–10, Dec. 2016.
[19] B. Mahdian and S. Saic, "Detection of copy–move forgery using a method based on blur moment invariants," *Forensic Sci. Int.*, vol. 171, nos. 2–3, pp. 180–189, Sep. 2007.
[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[21] M. Jaberi, G. Bebis, M. Hussain, and G. Muhammad, "Improving the detection and localization of duplicated regions in copy-move image forgery," in *Proc. 18th Int. Conf. Digit. Signal Process. (DSP)*, Fira, Greece, Jul. 2013, pp. 1–6.
[22] K. Li, H. Li, B. Yang, Q. Meng, and S. Luo, "Detection of image forgery based on improved PCA-SIFT," *Comput. Eng. Netw.*, vol. 277, pp. 679–686, Dec. 2014.
[23] V. Anand, M. F. Hashmi, and A. G. Keskar, "A copy move forgery detection to overcome sustained attacks using dyadic wavelet transform and SIFT methods," in *Proc. Asian Conf. Intell. Inf. Database Syst.*, Apr. 2014, pp. 530–542.
[24] J. Zheng, Y. Liu, J. Ren, T. Zhu, Y. Yan, and H. Yang, "Fusion of block and keypoints based approaches for effective copy-move image forgery detection," *Multidimensional Syst. Signal Process.*, vol. 27, no. 4, pp. 989–1005, Apr. 2016.
[25] X. Bo, W. Junwen, L. Guangjie, and D. Yuewei, "Image copy-move forgery detection based on SURF," in *Proc. Int. Conf. Multimedia Inf. Netw. Secur.*, Jiangsu, China, 2010, pp. 889–892.

[26] R. C. Pandey, S. K. Singh, K. K. Shukla, and R. Agrawal, "Fast and robust passive copy-move forgery detection using SURF and SIFT image features," in *Proc. 9th Int. Conf. Ind. Inf. Syst. (ICIIS)*, Gwalior, India, Dec. 2014, pp. 1–6.
[27] X.-Y. Wang, S. Li, Y.-N. Liu, Y. Niu, H.-Y. Yang, and Z.-L. Zhou, "A new keypoint-based copy-move forgery detection for small smooth regions," *Multimedia Tools Appl.*, vol. 76, no. 22, pp. 23353–23382, Nov. 2016.
[28] 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.
[29] Y. Li and J. Zhou, "Fast and effective image copy-move forgery detection via hierarchical feature point matching," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 5, pp. 1307–1322, May 2019.
[30] P. Mishra, N. Mishra, S. Sharma, and R. Patel, "Region duplication forgery detection technique based on SURF and HAC," *Sci. World J.*, vol. 2013, pp. 1–8, Nov. 2013.
[31] E. Ardizzone, A. Bruno, and G. Mazzola, "Copy–Move forgery detection by matching triangles of keypoints," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 10, pp. 2084–2094, Oct. 2015.
[32] C.-M. Pun, X.-C. Yuan, and X.-L. 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.
[33] Z. Shi, X. Shen, H. Kang, and Y. Lv, "Image manipulation detection and localization based on the dual-domain convolutional neural networks," *IEEE Access*, vol. 6, pp. 76437–76453, 2018.
[34] Y. Li, N. Liu, B. Zhang, K.-G. Yuan, and Y.-X. Yang, "Image multiple copy-move forgery detection algorithm based on reversed-generalized 2 Nearest-Neighbor," *J. Electron. Informat. Technol.*, vol. 37, pp. 1667–1673, Jul. 2015.
[35] 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.
[36] R. Toldo and A. Fusiello, "Robust multiple structures estimation with J-Linkage," in *Proc. Eur. Conf. Comput. Vision.*, Berlin, Germany, 2008, pp. 537–547.
[37] 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.
[38] M. Zandi, A. Mahmoudi-Aznaveh, and A. Talebpour, "Iterative copy-move forgery detection based on a new interest point detector," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 11, pp. 2499–2512, Nov. 2016.
[39] Y. Liu, Q. Guan, and X. Zhao, "Copy-move forgery detection based on convolutional kernel network," *Multimedia Tools Appl.*, vol. 77, no. 14, pp. 18269–18293, Nov. 2017.
[40] Y. Wu, W. Abd-Almageed, and P. Natarajan, "BusterNet: Detecting copy-move image forgery with Source/Target localization," in *Proc. ECCV Int.*, Oct. 2018, pp. 168–184.
[41] Y. Yin, L. Chen, Y. Xu, J. Wan, H. Zhang, and Z. Mai, "QoS prediction for service recommendation with deep feature learning in edge computing environment," *Mobile Netw. Appl.*, pp. 1–11, Apr. 2019.
[42] S. Pang, H. Chen, H. Liu, J. Yao, and M. Wang, "A deadlock resolution strategy based on spiking neural P systems," *J. Ambient Intell. Humanized Comput.*, pp. 1–12, Feb. 2019.
[43] T. Song, S. Pang, S. Hao, A. Rodríguez-Patón, and P. Zheng, "A parallel image skeletonizing method using spiking neural P systems with weights," *Neural Process. Lett.*, vol. 50, no. 2, pp. 1485–1502, Oct. 2018.

HAIPENG CHEN was born in 1978. Now, he is a Professor with the College of Computer Science and Technology, Jilin University, where he is also a member of the Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education. His research interests include image processing, pattern recognition, and multimedia information security.

XIWEN YANG was born in 1995. She received the B.E. degree in computer science and technology from the Shandong University of Science and Technology, China, in 2017. She is currently pursuing the M.E. degree with the College of Software, Jilin University. Her research interests include multimedia forensics and pattern recognition, especially on image copy-move detection and localization.

YINGDA LYU was born in 1983. She received the Ph.D. degree in computer science and technology from Jilin University, in 2015. Now, she is a Lecturer with the Center for Computer Fundamental Education, Jilin University. Her research interests include image processing, identification for image authenticity, and pattern recognition.