Accepted Manuscript

An improved method for SIFT-based copy-move forgery detection using non-maximum value suppression and optimized J-Linkage

Guonian Jin, Xiaoxia Wan

PII: S0923-5965(17)30094-2
DOI: http://dx.doi.org/10.1016/j.image.2017.05.010
Reference: IMAGE 15229

To appear in: *Signal Processing: Image Communication*

Received date : 13 January 2017
Revised date : 15 May 2017
Accepted date : 15 May 2017

Please cite this article as: G. Jin, X. Wan, An improved method for SIFT-based copy-move forgery detection using non-maximum value suppression and optimized J-Linkage, *Signal Processing: Image Communication* (2017), http://dx.doi.org/10.1016/j.image.2017.05.010

An improved method for SIFT-based copy-move forgery detection using non-maximum value suppression and optimized J-Linkage

Guonian Jin[a,b], **Xiaoxia Wan**[a,*]

[a] School of Printing and Packaging Engineering, Wuhan University, Wuhan, China

[b] College of Computer Science and Technology, Hubei University of Science and Technology, Xianning, China

Abstract. In looking to improve the detection performance of the keypoint-based method involving smooth tampered regions, there are three problems to be addressed, namely the nonuniform distribution of the keypoints, the discriminative power of low contrast keypoints, and the high computational cost of clustering. In this study, the classical implementation framework of the keypoint-based method is improved by introducing new techniques and algorithms in order to overcome these problems. First, to acquire uniformly distributed keypoints in the test image, we propose a new solution of selecting the keypoints by region instead of contrast. To this end, we first separate the keypoint detection and selection processes. After obtaining all discernible keypoints, we adapt the non-maximum value suppression algorithm to select keypoints by combining the contrast and density of each keypoint. Second, we apply the opponent scale-invariant feature transform descriptor to enhance the discriminative power of keypoints by adding color information. Finally, to alleviate the computational cost of clustering, we optimize the J-Linkage algorithm by altering the method of computing initial clusters and affine transformation hypotheses. For this purpose, we propose the matched pair grouping algorithm that can obtain a smaller number of initial clusters by utilizing the correspondence between the superpixels in the original and duplicated regions. Experiments performed on three representative datasets confirm that the proposed method can significantly improve the detection performance in smooth tampered regions, and considerably reduce the clustering time in the case of a mass of matched pairs, compared with the state-of-the-art methods.

Keywords: copy-move forgery detection; non-maximum value suppression; OpponentSIFT; simple linear iterative clustering; J-Linkage.

1. Introduction

In recent years, a number of tampered images have appeared in news reports, forensic evidence, and academic research, resulting in a negative impact on society. Copy-move forgery is a widely employed forging method, where one or more regions are copied and pasted to other locations within the same image [1-3]. In order to make such a forgery imperceptible, the duplicated region is often subjected to scaling, rotation, or blurring, which makes it more difficult to detect the copy-move forgery.

Detection methods utilizing feature matching techniques to find similar regions in the test image can be classified into two main categories: block-based and keypoint-based methods. Block-based methods divide the test image into overlapping blocks, and use different feature algorithms to characterize every block, thus enhancing the robustness to additive noise, JPEG lossy compression, brightness adjustment, and so on. However, there remain problems with such methods, such as high computational cost and poor affine transformation invariance [4]. Keypoint-based methods can effectively overcome these problems. Keypoint-based methods [4-6] first perform keypoint detection and feature matching, and then cluster the matched pairs (e.g., with J-Linkage) and estimate the affine transformation. Finally, tampered regions are detected using the region correlation map. These methods achieve an excellent detection performance, especially in textural tampered regions.

In attempting to improve the detection performance of the keypoint-based method involving smooth tampered regions, there are three challenges. The first challenge is to increase the uniform distribution density of the keypoints in the test image. Because the original and duplicated regions can be located anywhere in the test image, high contrast keypoints are insufficient to cover all possible tampered regions. This is also the reason why keypoint-based methods fail in smooth tampered regions. The second challenge is to enhance the

* Address all correspondence to: Xiaoxia Wan, E-mail: wan_whu@hotmail.com.

discriminative power of low contrast keypoints. When achieving the uniformly distributed keypoints across the test image, there are sure to be many low contrast keypoints. Hence, enhancing the discriminative power of low contrast keypoints would be beneficial in improving the matching performance. The final challenge is to reduce the computing time required for clustering. When we focus on detecting smooth tampered regions, there are a large number of matched pairs in many test images. Furthermore, these matched pairs are useful for improving the detection performance. However, the time complexity of the J-Linkage algorithm is quadratic on the number of matched pairs [7].

In this study, we improve the classical implementation framework of the keypoint-based method to overcome these challenges. First, we propose a new solution of selecting keypoints by region instead of contrast. To this end, we separate the keypoint detection and selection processes. In the keypoint detection process, we collect all discernible keypoints. In the selection process, we adapt the non-maximum value suppression (NMS) algorithm to ensure that the keypoints are uniformly distributed in the test image, and have the highest contrast in local region. Second, in order to enhance the discriminative power of low contrast keypoints, we adopt the opponent scale-invariant feature transform (OpponentSIFT) descriptor to extract the feature vector of the keypoints. Finally, we optimize the J-Linkage algorithm by altering the method of computing the initial clusters and affine transformation hypotheses. We segment test image into superpixels, and group the matched pairs to obtain the initial clusters and affine transformation hypotheses of the J-Linkage algorithm. In this manner, the clustering time is significantly reduced in the case of a mass of matched pairs.

The remainder of this paper is organized as follows. In Section 2, existing copy-move forgery detection methods are briefly reviewed. The proposed method is presented in Section 3. The experimental results and comparisons are discussed in Section 4. Finally, our conclusions are given in Section 5.

2. Related work

As mentioned above, the existing methods for copy-move forgery detection can be classified into two main categories: block-based and keypoint-based methods.

2.1. Block-based methods

Fridrich et al. [8] first investigated the problem of copy-move forgery, and proposed the block-based method. They tiled the test image with overlapping blocks of a fixed size, and employed the discrete cosine transform (DCT) to characterize each block. Then, in the matching stage lexicographic sorting was exploited to overcome the computational complexity. Many robust detection algorithms have proposed on the basis of this method, such as the intensity-based method [9], frequency domain-based method [10], and moment-based method [11]. To reduce the computing time for the matching stage, Cozzolino et al. [12] proposed a method based on the PatchMatch algorithm. Although these methods improve the detection performance, there remain some shortcomings, such as high computational cost and poor affine transformation invariance.

2.2. Keypoint-based methods

Keypoint features are robust to scaling, rotation, occlusion, and so on, making them well-suited to copy-move forgery detection. The scale-invariant feature transform (SIFT) was first applied to forgery detection by Huang et al. [13]. Pan et al. [14] used the random sample consensus (RANSAC) algorithm to estimate the affine transformation matrix between the original and duplicated regions, and detected the duplicated regions using the region correlation map. Amerini et al. [5] proposed the generalized 2-nearest neighbor (g2NN) matching process, and used the agglomerative hierarchical clustering to cluster matched pairs, which effectively solved the detection problem of multiple cloned regions. Christlein et al. [4] integrated the above two methods to implement the overall framework of the keypoint-based method, which consisted of feature extraction and keypoint matching, the clustering of matches and estimation of affine transformations, and the localization of duplicated regions.

Existing research in this field mainly focuses on improving the feature extraction and clustering stages.

For textural tampered regions, existing improvements have been proposed in the following three aspects. First, alternative types of keypoint and feature descriptor have been applied, such as speeded up robust feature

(SURF) [15]. Second, some methods have combined different kinds of keypoint and feature descriptor, such as the Harris corner and SIFT descriptor employed in [16]. Third, the keypoints and feature vectors can be extracted in different color spaces, such as the method of extracting the SURF feature in the opponent color space in [17]. Regarding smooth tampered regions, Guo et al. [18] employed the adaptive non-maximal suppression algorithm (ANMS) to select Harris corners, and extracted DAISY descriptors. Yu et al. [19] adopted a two-stage detection method to detect Harris corners, and extracted multi-support region order-based gradient histogram (MROGH) and hue histogram (HH) descriptors. Both of these methods have poor robustness to scaling, owing to the adoption of the Harris corner.

In the clustering stage, Amerini et al. [5] utilized the spatial coordinates of the keypoints to implement agglomerative hierarchical clustering. Because only the coordinates of the matched pairs are taken into account, and the matching constraint between points is ignored, this method fails when the duplicated region is spatially close to the original region. The authors eventually proposed an improved method to overcome this shortcoming by using the J-Linkage algorithm [6]. Because the time complexity of the J-Linkage algorithm is quadratic on the number of matched pairs, the clustering time increases significantly as the number of matched pairs increases.

In the last two years, some methods have been proposed that integrate block-based and keypoint-based methods. Silva et al. [20] proposed a method based on multi-scale analysis and voting processes, which determined the range of suspicious regions using SURF feature matching and clustering, and then applied the block-based method to detect tampered regions in the multi-scale space. Pun et al. [21] applied adaptive over-segmentation and the forgery region extraction algorithm to detect tampered regions. Li et al. [22] also employed the image segmentation method. After detecting suspected tampered regions, the iterative nearest neighbor algorithm was adopted to gradually improve the detection precision. However, the computational cost of this method is prohibitive for large images.

As the focus of research has shifted from textural tampered regions to smooth tampered regions, a method that gives uniformly distributed keypoints with high density in the image is urgently required to achieve a strong detection performance. In addition, this will further increase the computing time required for the clustering stage. In the next section, we propose an improvement to the classical implementation framework of keypoint-based methods [4, 6], in order to solve these problems.

3. Proposed method

In this paper, we propose an improved classical implementation framework for the keypoint-based method by introducing new techniques and algorithms, as illustrated in Fig. 1. First, in order to improve the keypoint detection process, we propose a new solution of selecting the keypoints by region instead of contrast. Second, we extract OpponentSIFT descriptors in order to enhance the discriminative power of the keypoints. Finally, we optimize the J-Linkage algorithm by means of image segmentation and the matched pair grouping algorithm.

3.1. Keypoint detection and feature matching

Existing keypoint-based methods are not sufficient for the new application scenario in which the focus is on detecting smooth tampered regions [12]. They do not take into account the different keypoint requirements between copy-move forgery detection and the computer vision applications, such as 3D reconstruction and motion tracking. Computer vision applications require only high contrast keypoints, and so low contrast keypoints are automatically filtered out according to a preset contrast threshold during keypoint detection. As a result, there are scarcely any keypoints in the smooth region, as shown in Fig. 2(b). However, because tampered regions can be located anywhere in the test image, copy-move forgery detection requires uniformly distributed keypoints across the test image, and these high contrast keypoints are insufficient to cover all possible tampered regions. Hence, the reason for the failure of keypoint-based method is not the insufficient keypoints in the test image, but rather the unsuitable manner of selecting the keypoints.

In order to improve the detection performance of the keypoint-based method involving smooth tampered regions, we propose a new solution of selecting the keypoints by region instead of contrast. To this end, we separate the keypoint detection and selection processes. In the keypoint detection process, we obtain all

discernible keypoints in the test image, as shown in Fig. 2(c). Then, in the keypoint selection process, we adapt the NMS algorithm to select keypoints by combining the contrast and density of each keypoint. Moreover, we apply the OpponentSIFT descriptor to enhance the discriminative power of low contrast keypoints.

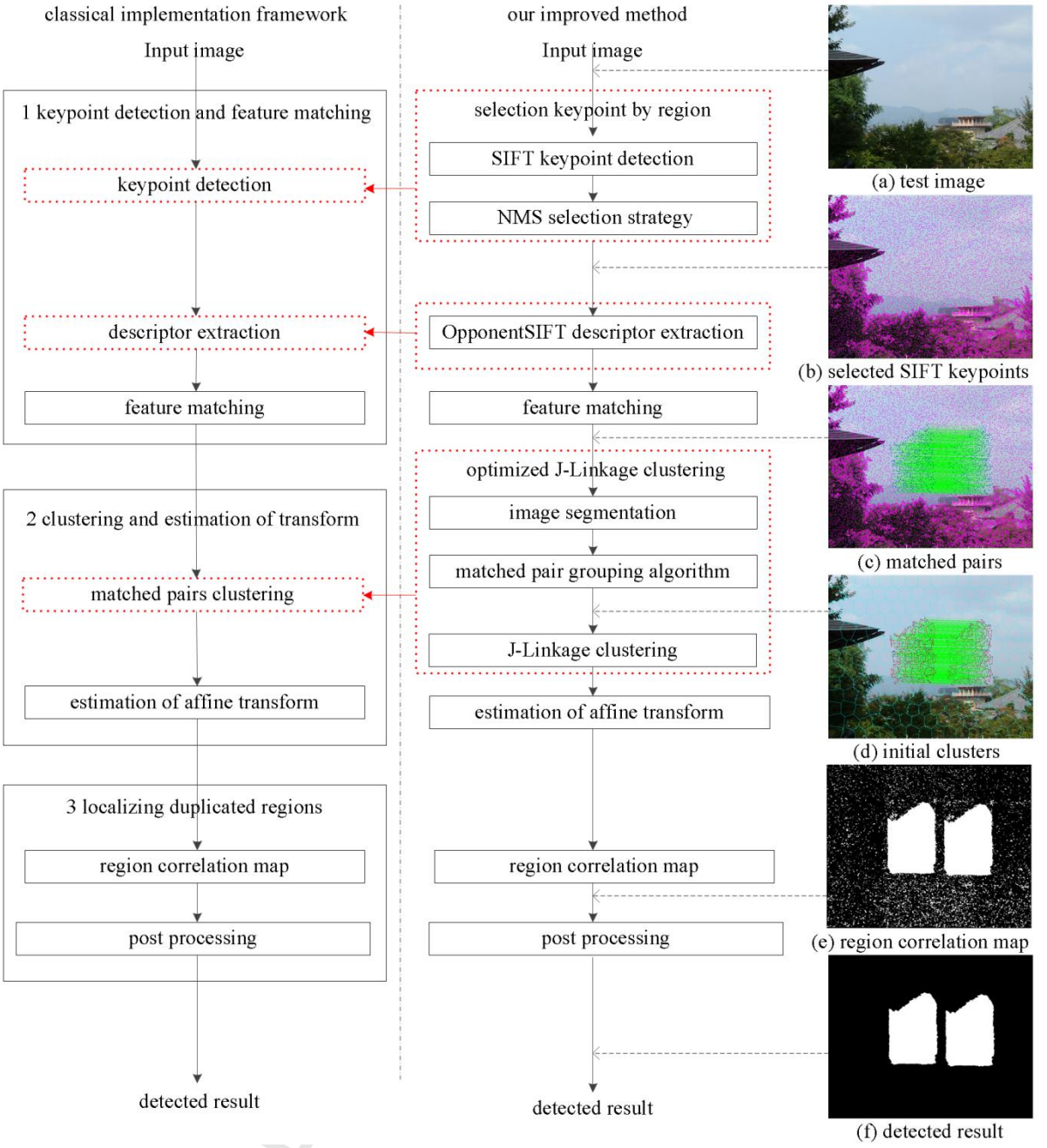


Fig. 1. Main steps of the proposed method.

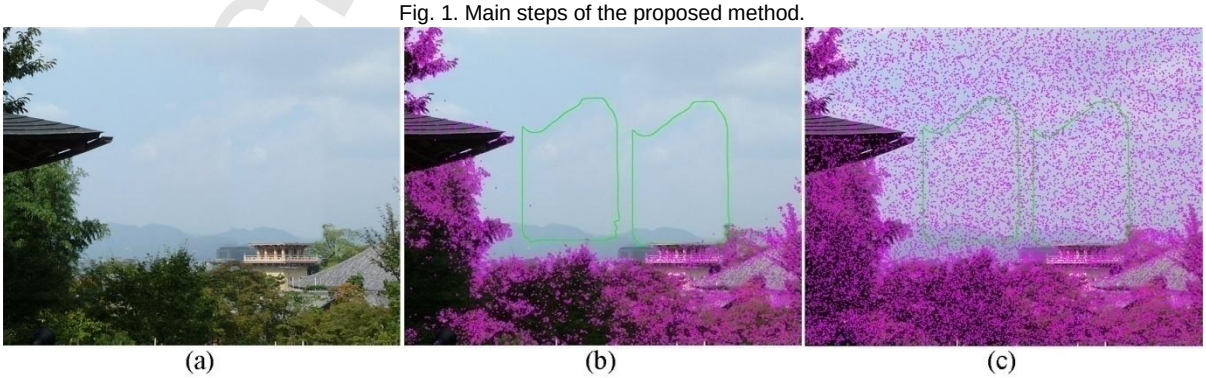


Fig. 2. Examples of keypoint detection: (a) test image, (b) high contrast keypoints, (c) all discernible keypoints.

In the following subsections, we first collect all discernible keypoints during the keypoint detection process. Next, we adapt the NMS algorithm to acquire uniformly distributed keypoints in the test image. Then, we convert the test image from RGB to the opponent color space, and extract the OpponentSIFT descriptors. Finally, we use the g2NN method to match the features.

3.1.1. SIFT keypoint detection

We first study the mechanism of selecting keypoints during the detection of SIFT keypoints. Based on this, we collect all discernible keypoints in the test image in order to address the problem of insufficient keypoints.

SIFT keypoints represent the stable local extrema positions in scale space. When an extremum point is sought in the scale space, the interpolated location of the maximum is determined by fitting a 3D quadratic function to the local sample points [23]. The Taylor expansion (up to the quadratic terms) is applied to the scale-space function D(x), by regarding the sample point as the origin. D(x) can be expressed as

$$\mathrm{D}(\mathrm{x})=\mathrm{D}+\frac{\partial \mathrm{D}^{\mathrm{T}}}{\partial \mathrm{x}}\mathrm{x}+\frac{1}{2}\mathrm{x}^{\mathrm{T}}\frac{\partial^2 \mathrm{D}}{\partial \mathrm{x}^2}\mathrm{x}\ . \tag{1}$$

The location of the extremum, $\hat{x}$, is determined by taking the derivative of this function with respect to x and setting it to zero, giving

$$\hat{\mathrm{x}}=-\frac{\partial^2 \mathrm{D}^{-1}}{\partial \mathrm{x}^2}\frac{\partial \mathrm{D}}{\partial \mathrm{x}}\ . \tag{2}$$

If $\hat{x}$ is greater than 0.5 in any dimension, then the sample point is changed, and the location of the extremum is solved further in an iterative manner. The function value $\mathrm{D}(\hat{\mathrm{x}})$ at this extremum is useful for rejecting unstable extrema with low contrast, which can be expressed as

$$\mathrm{D}(\hat{\mathrm{x}})=\mathrm{D}+\frac{1}{2}\frac{\partial \mathrm{D}^{\mathrm{T}}}{\partial \mathrm{x}}\hat{\mathrm{x}}. \tag{3}$$

After the value of the function $\mathrm{D}(\hat{\mathrm{x}})$ is obtained, the keypoints with high contrast are selected by imposing a preset contrast threshold. All extrema with a value of $|\mathrm{D}(\hat{\mathrm{x}})|$ less than this preset contrast threshold are discarded.

By analyzing the selection process of SIFT keypoints, we find that the keypoints are usually selected by imposing a preset contrast threshold during the keypoint detection. This process fully satisfies the requirements of computer vision applications, because they only require high contrast keypoints. However, because copy-move forgery detection requires enough keypoints to cover the entire image, we collect all discernible keypoints in this process. In other words, we set this contrast threshold to 0.

3.1.2. NMS selection strategy

In this subsection, we describe the selection of keypoints by region. We obtain a large number of discernible keypoints in the detection process. However, for copy-move forgery detection, there will undoubtedly be a certain number of redundant keypoints, which will increase the computational cost of feature extraction and feature matching. Therefore, it is necessary to reduce the number of redundant keypoints on the condition of still ensuring a high detection performance. Extensive experiments show that we can reduce the keypoints by around 50% without affecting the detection performance.

We adopt the NMS algorithm [24] as the keypoint selection strategy. The NMS algorithm first appeared in the Canny algorithm, which is aimed at searching for local maxima and suppressing non-maxima elements. For a specific suppression radius r, the NMS algorithm only retains the highest contrast keypoint in each (2r+1)×(2r+1) region. In this manner, the keypoints have a considerably more uniform spatial distribution across the test image, which is highly suited to copy-move forgery detection, as illustrated in Fig. 3(c). In contrast, if we select the keypoints by contrast, then there will be scarcely any keypoints in the smooth tampered regions if the same number of keypoints are selected, as shown in Fig. 3(d).

Moreover, we give special consideration to high contrast keypoints and sparse keypoint regions. In our method, the keypoints with contrast not less than 0.0133 are considered as high contrast keypoints. Namely, these are the default SIFT keypoints in OpenCV. On the other hand, regions where the number of keypoints is lower than the average are taken as sparse keypoint regions. For these two cases, we halve the suppression radius r. The keypoint selection algorithm based on contrast and density is presented in Algorithm 1. Through extensive experiments, we determined that when the suppression radius r is set to 5 a good balance is achieved

between the detection performance and computing time. The impact of the suppression radius is investigated further in the experimental section.

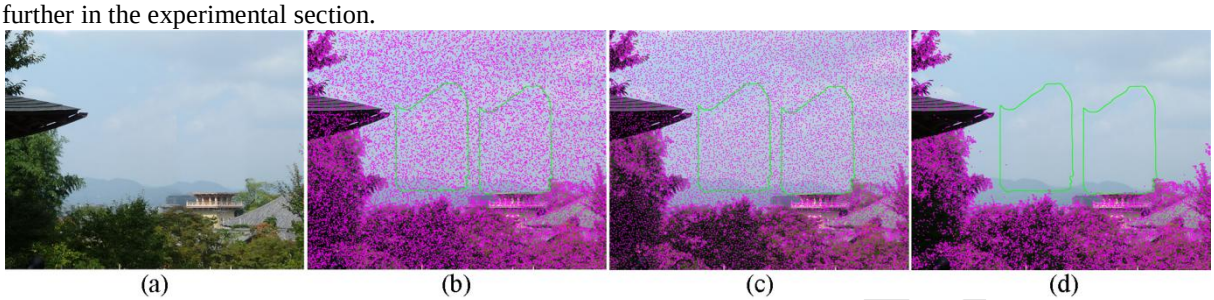


Fig. 3. Keypoint selection: (a) test image, (b) all discernible keypoints, (c) keypoints selected by NMS, (d) keypoints selected by contrast.

Algorithm 1. Keypoint selection algorithm based on contrast and density

Input: keypoint set *K*, suppression radius *R*, contrast threshold *T*.

Output: selected keypoint set *K*'.

Compute the number of keypoints for each pixel, stored in *KPC(y, x)*.
Compute the keypoint density *KPD(y, x)* for each pixel using the integral image, $KPD(y,x) = \sum_{m=-2R}^{2R}\sum_{n=-2R}^{2R} \mathrm{KPC}(y+m, x+n)$.
Compute the average of the keypoint density, stored in *AvgD*.
Compute the mapping matrix *CoordKP* for all keypoints.
for each row *y* in *CoordKP*
    for each column *x* in *CoordKP*
        If there is no keypoint in the current pixel (y, x), then next loop.
        Set the suppression radius r=R.
        Adjust the suppression radius *r* according to the keypoint contrast or the keypoint density. If the keypoint contrast is not less than the threshold *T* or the keypoint density is less than the average density *AvgD*, the suppression radius *r* is adjusted to R/2.
        Select keypoints using the suppression radius *r*. The current keypoint is retained if it is the highest contrast keypoint in the (2r+1)×(2r+1) region.
    end for
end for
Assign all keypoints in *CoordKP* to *K*'.

3.1.3. OpponentSIFT descriptor extraction

We adopt OpponentSIFT [25] to compute the color descriptor for each keypoint. Existing SIFT-based methods extract SIFT descriptors because they use high contrast SIFT keypoints, which have better discriminative power. However, in our method, many low contrast keypoints are required to achieve a uniform distribution of keypoints across the test image. Therefore, it is necessary to enhance the discriminative power of these keypoints. The literature [25] indicates that color plays an important role in distinguishing different objects, and OpponentSIFT descriptor can effectively improve the discriminative power of SIFT keypoint, owing to the combination of intensity and color descriptors. Based on the results of this study, we apply the OpponentSIFT descriptor to extract the feature vector of the keypoints. The effectiveness of the OpponentSIFT descriptor is explored further in the experimental section.

The OpponentSIFT descriptor describes all of the channels in the opponent color space using the SIFT descriptor. To this end, the color image is converted from the RGB color space to the opponent color space [25]. The conversion formula is given as

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{3}} \end{pmatrix}, \quad (4)$$

where O_1 and O_2 represent the color information, while O_3 is equal to the intensity information. Then, the SIFT descriptors are extracted from these three channels, and concatenated to obtain the OpponentSIFT descriptor.

3.1.4. Feature matching

After obtaining the OpponentSIFT descriptors, we can roughly determine whether there are duplicated regions in the test image via feature matching. In copy-move forgery, a tampered image generally contains two or more duplicated regions, and so the keypoints in these regions have similar descriptor vectors. In the feature matching stage, we adopt the g2NN matching process proposed in [5], which effectively solves the detection problem of multiple cloned regions.

For the sake of clarity, a matched pair $p = (s, s')$, s is referred to as a source keypoint, and s' is called a corresponding keypoint.

3.2. Optimized J-Linkage clustering

After obtaining the matched pairs, we can cluster these matched pairs using the J-Linkage algorithm to estimate more accurate affine transformation matrices and solve the problem of detecting multiple cloned regions. In J-Linkage clustering, a random sampling is first performed on matched pairs to generate M affine transformation hypotheses. Subsequently, each matched pair is assigned to an initial cluster, and these initial clusters undergo an agglomerative clustering process in conceptual space. The time complexity of these two steps is quadratic on the number of matched pairs. Hence, the clustering time increases significantly as the number of matched pairs increases. When we focus on detecting smooth tampered regions, there are a large number of matched pairs in many test images, which are useful for improving the detection performance. In this context, the clustering time of the J-Linkage algorithm becomes a serious problem.

To address this problem, we group the matched pairs by utilizing the characteristics of the copy-move tampered image in order to obtain a smaller number of initial clusters. That is to say, we exploit the correspondence between the original and the duplicated regions. By segmenting the tampered images, we find that the original and duplicated regions are segmented in a similar manner, and there is a good correspondence between the superpixels in the original and duplicated regions, as shown in Fig. 4(b). The correspondence between the superpixels fully satisfies the requirements for grouping the matched pairs, such as S1 and D1, S2 and D2, etc. Based on this, we first segment the test image into superpixels, and then we group the matched pairs according to this correspondence, as shown in Fig. 4(c). In this manner, we can obtain a smaller number of initial clusters compared with the existing J-Linkage algorithm, where each matched pair is assigned to an initial cluster. In the example of Fig. 4, there are 4826 matched pairs, but we only acquire 237 initial clusters. Furthermore, we can estimate the affine transformation matrices according to these initial clusters as the affine transformation hypotheses of the J-Linkage algorithm.

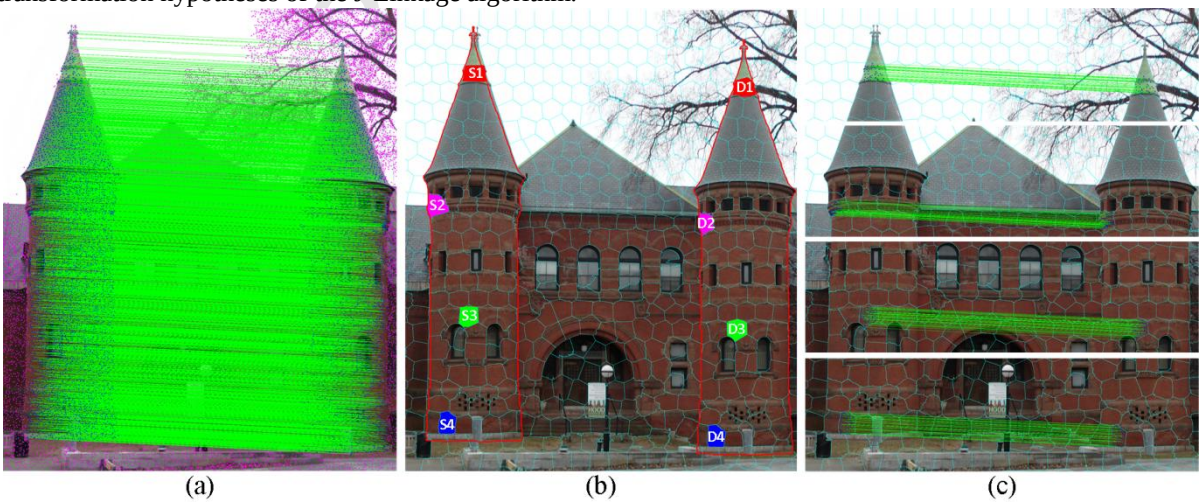


Fig. 4. Computing initial clusters: (a) matched pairs, (b) image segmentation and corresponding superpixels, (c) examples of initial clusters.

In the next subsections, we first segment the test image into meaningful superpixels. Next, we obtain the initial clusters of the J-Linkage algorithm using the matched pair grouping algorithm, and we compute the affine transformation hypotheses of the J-Linkage algorithm. Finally, we perform the J-Linkage clustering, and estimate the affine transformation matrix using the RANSAC algorithm.

3.2.1. Image segmentation

In our method, we apply the simple linear iterative clustering (SLIC) algorithm [26] to segment the test image into superpixels that consist of perceptually meaningful regions and have well-defined boundaries for these duplicated regions. The only parameter of the SLIC algorithm is the desired number of superpixels, which in our method is computed as the ratio of the size of the image to the size of the superpixel. Because we only apply the correspondence between the superpixels in the original and duplicated regions to group the matched pairs, which does not require these two superpixels to be exactly the same (e.g., Fig. 4(b)), the size of the superpixels is not the main factor affecting the detection performance. Extensive experiments show that as long as the superpixel size is moderate, such as from 60×60 to 140×140, it has no effect on the detection results. In our experiment, the size of the superpixels is set to 100×100, and for small images the number of superpixels is set to at least 50.

Although the method in [21, 22] uses the SLIC algorithm, our purpose is completely different. That method first segments the test image into superpixels using the SLIC algorithm and extracts the keypoints from each superpixel. Then, suspicious regions can be detected by matching the keypoints between superpixels. Unlike in that case, we compute the initial clusters and the affine transformation hypotheses of the J-Linkage algorithm by utilizing the correspondence between the superpixels obtained by the SLIC algorithm. In other words, we optimize the J-Linkage algorithm with the help of the image segmentation.

As mentioned above, we only optimize the J-Linkage algorithm in the case of a mass of matched pairs. Because the image segmentation requires a certain amount of computing time, the segmenting time is greater than the clustering time if the number of matching pairs is small. Experiments show that when the number of matched pairs exceeds 600, our proposed method has the advantage in terms of computing time. Therefore, in our method, when the number of matching pairs is lower than 600 the J-Linkage algorithm is applied directly.

3.2.2. Matched pair grouping algorithm

In this subsection, we describe how to compute the initial clusters of the J-Linkage algorithm. First, each superpixel is assigned as an initial cluster. Then, the matched pairs are assigned to these initial clusters, based on the location of the source keypoint and the matching constraints between the keypoints. According to the relationship between the matched pairs and the superpixels, there are five cases for assigning the matched pairs, as shown in Fig. 5. For two different matched pairs, the source keypoints may be the same keypoint (e.g., Fig. 5(b, d)), or they may be different keypoints (e.g., Fig. 5(c, e)). In the same way, the corresponding keypoints may lie in the same superpixel (e.g., Fig. 5(b, c)) or in different superpixels (e.g., Fig. 5(d, e)). In addition, there is one special case, as shown in Fig. 5(a). The matched pair grouping algorithm is presented in Algorithm 2.

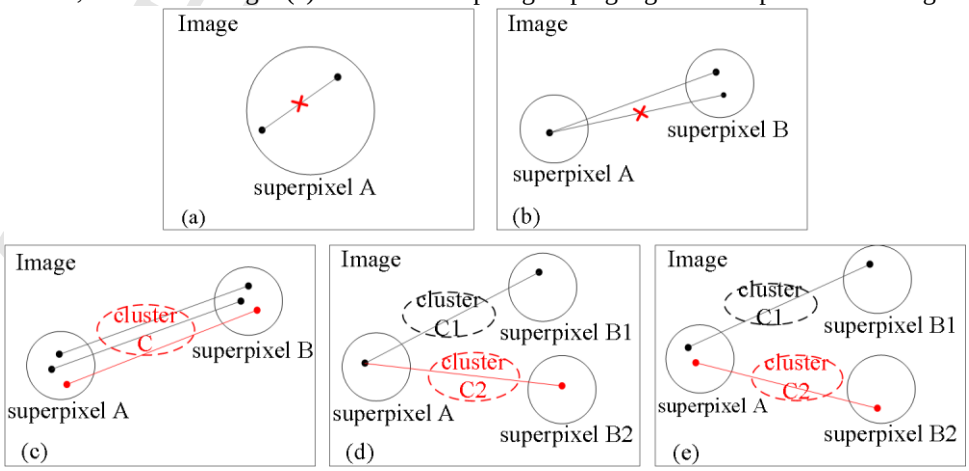


Fig. 5. Five assignment cases: (a) source keypoint and corresponding keypoint in the same superpixel, (b) multiple corresponding keypoints of a source keypoint in the same superpixel, (c) a cluster of matched pairs, (d) multiple corresponding keypoints of a source keypoint in different superpixels, (e) source keypoints in the same superpixel but corresponding keypoints in different superpixels.

Algorithm 2. Matched pair grouping algorithm

Input: matched pairs *P*; superpixels *SP*.

Output: initial clusters *C*.

1: Create an array *X* according to the superpixels *SP*, in which each element is also an array and save the clusters for this superpixel.
2: Group matched pairs P by the coordinate of the source keypoint, named *GP*.
3: **for** each subgroup GP_i in *GP*
4: Remove the matched pairs where the source keypoint and the corresponding keypoint are located inside the same superpixel, as shown in Fig. 5(a).
5: Retain the matched pair in which the corresponding keypoint has the highest contrast. If a source keypoint has multiple corresponding keypoints and they are located inside the same superpixel, as shown in Fig. 5(b).
6: **for** each matched pair p_j in subgroup GP_i
7: Obtain the cluster list *CL* according to the superpixel where the source keypoint of p_j is located.
8: Search the cluster C_t that fits with p_j in *CL*. In other words, their source keypoints and corresponding keypoints are located respectively inside two corresponding superpixels.
9: If found, p_j is added to C_t, as shown in Fig. 5(c).
10: If not found, p_j is added to a new cluster, and this new cluster is inserted into *CL*, as shown in Fig. 5(d, e).
11: **end for**
12: **end for**
13: Assign all clusters in *X* to *C*.

In copy-move forgery, because there is a correspondence between the superpixels in the original and duplicated regions, all matched pairs in which the source keypoint and corresponding keypoint are respectively located inside two corresponding superpixels are grouped into an initial cluster, as shown in Fig. 5(c). A keypoint has multiple corresponding keypoints in the case of multiple cloned regions. Therefore, a new cluster is assigned to each of these matched pairs, as shown in Fig. 5(d).

Because of the local characteristics of the image, there are highly similar keypoints in adjacent regions. Existing detection methods adopt the shortest distance threshold to eliminate these matched pairs. The threshold is set to 50 empirically, instead of determining a appropriate value based on the information of the image itself, such as textural details. In this study, we effectively solve this problem, as illustrated in Fig. 5(a).

Once the initial clusters are obtained, those in which the number of matched pairs is not lower than 3 are selected for estimating the affine transformation matrices as the affine transformation hypotheses of J-Linkage algorithm. We adapt the SATS algorithm [27] to estimate the affine transformation matrix.

3.2.3. Clustering and estimation of transformation matrix

After obtaining the initial clusters and affine transformation hypotheses, the initial clusters are clustered using the J-Linkage process. The preference set vector of an initial cluster is first computed as the intersection of the preference sets of its matched pairs. For each pair p in an initial cluster, a preference set vector PS(p) is defined to indicate the preferred affine transformation hypotheses for this matched pair. Given M hypotheses $\mathrm{T} = \{\mathrm{T}_1, \ldots, \mathrm{T}_\mathrm{M}\}$, the preference set vector PS(p) is defined as

$$\mathrm{PS}(\mathrm{p}) = \{\mathrm{PS}_1(\mathrm{p}), \ldots, \mathrm{PS}_\mathrm{M}(\mathrm{p})\},\ \mathrm{PS}_\mathrm{i}(\mathrm{p}) = \begin{cases} 1 & \text{if p is an inliner of } \mathrm{T_i}, \\ 0 & \text{otherwise}. \end{cases} \quad (5)$$

This means that the distance between the hypothesis T_i and the matched pair p is less than a preset threshold (3 in our experiment). Because the initial clusters in the duplicated regions have similar affine transformations, they will have similar conceptual representations. In other words, they will cluster in the conceptual space $\{0,1\}^\mathrm{M}$.

In the J-Linkage clustering process, for each step of the algorithm the two clusters with shortest distance in the conceptual space are merged. The preference set vector of the new cluster is computed as the intersection of the preference sets of the two clusters, and the distance between two clusters is computed as the Jaccard distance between the respective preference sets [7]. Given two sets A and B, the Jaccard distance is defined as

$$J_\delta(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}. \quad (6)$$

The Jaccard distance measures the degree of overlap of two sets, and ranges from 0 (identical sets) to 1 (disjoint sets). Thus, J-Linkage clustering ends when the Jaccard distance between all clusters is 1.

After clustering, all of the clusters in which the number of matched pairs is lower than 4 are discarded. Then, the RANSAC algorithm [28] is adopted to obtain the affine transformation matrix of each cluster.

3.3. Localizing duplicated regions

In this section, we describe how the duplicated regions can be determined using the region correlation map and post processing.

3.3.1. Computing region correlation map

In order to localize duplicated regions, we first compute the correlation map $C_f(x)$ [4, 6] between the original image I and the warped image W, using the estimated affine transform matrix H and the inverse transformation H^{-1} respectively. The correlation map $C_f(x)$ is defined as

$$C_f(x) = \frac{\sum_{v \in \Omega(x)} (I(v) - \bar{I})(W(v) - \bar{W})}{\sqrt{\sum_{v \in \Omega(x)} (I(v) - \bar{I})^2 (W(v) - \bar{W})^2}}. \quad (7)$$

Here, $\Omega(x)$ is a 7 pixels neighboring area centered at each pixel x; I(v) and W(v) denote the pixel intensities at the location v; $\bar{I}$ and $\bar{W}$ are the average pixel intensities of I and W in the neighboring area $\Omega(x)$.

3.3.2. Post processing

After obtaining the region correlation map, we identify duplicated regions through the following four steps. First, we employ Gaussian filtering and thresholding algorithm to create a binary image. In our experiment, the Gaussian kernel size is 7 pixels, and the binary threshold is 0.55. Next, we apply the morphological closing operation to smooth the boundaries of the detected regions, and adopt the morphological hole filling algorithm [29] to fill the holes in the detected regions. The element shape of closing operation is a rectangular structuring element, which is of size 3×3 with the origin in the center. Subsequently, we discard all small isolated regions that have areas of less than 0.05% of the image. Finally, we verify the detected regions using the matched pairs to ensure the correctness of the detected regions.

4. Experimental results

In this section, we conduct a series of experiments to validate the proposed method. First, we analyze the influence of the suppression radius parameter on the detection performance and computing time. Next, we evaluate the effectiveness of the NMS algorithm and OpponentSIFT descriptor in our method. Then, we evaluate the effectiveness of the optimized J-Linkage algorithm. Finally, we compare the proposed method with two state-of-the-art methods, which also apply keypoint detection and feature matching techniques, but use different methods to localize the duplicated regions.

For the sake of convenience, a label is associated to each chosen method, as listed in Table 1. The key parameters related to our method are listed in Table 2.

Table 1 Label associated with each method

Method	Label
Christlein et al. [4]	AHC
Amerini et al. [6]	JLC
Silva et al. [20]	SILVA
Li et al. [22]	SegmentBased

Table 2 The parameter setting of the proposed method

Parameters	Description
cth = 0	Contrast threshold for keypoint detection, in Section 3.1.1.
r = 5	Suppression radius for keypoint selection, in Section 3.1.2.
T = 0.6	Ratio threshold in g2NN matching process, in Section 3.1.4.
Size = 100×100	Superpixel size threshold for image segmentation, in Section 3.2.1.
D = 3	Distance threshold of inliers in J-Linkage algorithm, in Section 3.2.3.
th = 0.55	Binary threshold for the correlation map, in Section 3.3.2.
A = 0.05%	Minimum area threshold for the tampered region, in Section 3.3.2.

4.1. Datasets

Our method is evaluated on three public datasets. The first dataset, GRIP, was provided by Cozzolino et al. [12]. It consists of 80 images, each with 768×1024 pixels. Most of the images are smooth, which presents a considerable challenge to keypoint-based methods.

The second dataset, FAU, was constructed by Christlein et al. [4]. It includes 48 base images, and 87 copied snippets. The average size of the images is about 3000×2300 pixels. The copied snippets are from the categories of living, nature, man-made and mixed, and they range from overly smooth to highly textured. Copy-move forgeries can be created by copying, scaling, or rotating these semantically meaningful snippets.

The third dataset, MICC-F600, was introduced by Amerini et al. [6]. It contains 440 original images and 160 forged images. The images have different resolutions ranging from 800×533 to 3888×2592 pixels. The 160 tampered images are subjected to four types of tampering operations, such as translation, multiple cloned, rotation, and rotation and scaling.

4.2. Evaluation metrics

In our paper, the precision and recall are used to evaluate the detection performance of the proposed method. Precision is defined as the ratio of the number of correctly detected tampered pixels to the number of totally detected tampered pixels, as stated in (8). Recall is defined as the ratio of the number of correctly detected tampered pixels to the number of tampered pixels in the ground-truth image, as stated in (9). In additional, another criterion F_1 that combines both precision and recall is also computed, as stated in (10).

$$\text{precision} = \frac{|\{\text{CMF pixels}\} \cap \{\text{detected CMF pixels}\}|}{|\text{detected CMF pixels}|} \tag{8}$$

$$\text{recall} = \frac{|\{\text{CMF pixels}\} \cap \{\text{detected CMF pixels}\}|}{|\text{CMF pixels}|} \tag{9}$$

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{10}$$

4.3. Analysis of suppression radius parameter

In this subsection, we analyze the influence of the suppression radius *r* on the detection performance and computing time. As previously described, to obtain uniformly distributed keypoints across the test image, we collect all discernible keypoints in the keypoint detection process, which undoubtedly results in a number of redundant keypoints for copy-move forgery detection. These redundant keypoints will increase the computational cost of feature extraction and feature matching. Therefore, we try our best to reduce the number of redundant keypoints using the NMS algorithm, whilst still preserving the detection performance. Hence, the choice of the suppression radius r clearly affects the detection performance and computing time.

Because the influence of the suppression radius on the detection performance is related to the size of the tampered region, we select the 15 images with the smallest tampered regions from the dataset FAU, and resize the cloned regions to 80%. The suppression radius *r* varies from 0 to 9, and the test results are shown in Fig. 6. The computing time consists of the feature extraction time and feature matching time, because this is where the redundant keypoints mainly increase the computational cost. According to the test results, the detection performance is relatively stable when the suppression radius *r* is between 0 and 5. When the suppression radius is 6, detection failures begin to appear. In contrast, the computational cost of redundant keypoints is rapidly

reduced as the suppression radius r increases from 2 to 5. Therefore, we obtain a good trade-off between the detection performance and computing time by setting $r = 5$.

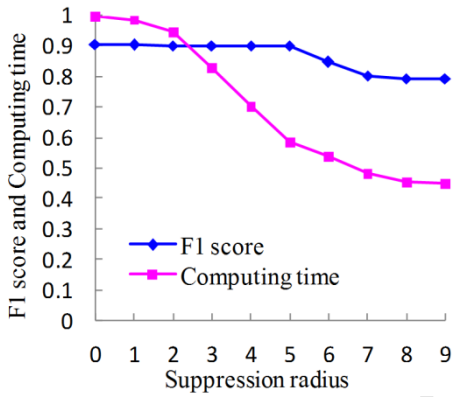


Fig. 6. Influence of suppression radius on detection performance and computing time.

4.4. Evaluation of NMS algorithm and OpponentSIFT descriptor on GRIP

The main objective of this subsection is to evaluate the effectiveness of our measures to improve the keypoint detection and feature matching stage. In the dataset GRIP, most of images are highly smooth, and there have fewer keypoints if we detect the default SIFT keypoints in OpenCV. Therefore, this is highly suitable for the purpose of our test. We first evaluate the effectiveness of the NMS algorithm and OpponentSIFT descriptor by comparison with the JLC method. Then, we further evaluate the influence of the OpponentSIFT descriptor.

4.4.1. Effect of NMS algorithm and OpponentSIFT descriptor

In this test, we validate the effect of the NMS algorithm and OpponentSIFT descriptor. To analyze the experimental results more clearly, we classify the test images into two categories. The first type contains 70 plain tampered images. Among them, there are 28 tampered images involving very smooth tampered regions. The other type contains 10 tampered images involving similar but genuine regions (SGRs). The test results are presented in Table 3.

Table 3 Comparison results with the JLC method

Method		Precision	Recall	F_1
JLC		0.558	0.622	0.588
NMS + SIFT		0.866	0.920	0.892
NMS + OpponentSIFT	70 plain tampered images	0.957	0.936	0.946
	10 SGRs tampered images	0.500	0.966	0.659
	Average	0.900	0.939	0.919

For the 70 tampered images of the first type, our proposed method detects 69 tampered images. Although there are 28 tampered images involving highly smooth tampered regions, Table 3 shows that our proposed method still achieves a good detection performance. As mentioned above, the JLC method selects high contrast keypoints by imposing a preset contrast threshold during the detection of keypoints, and so there are scarcely any keypoints in the smooth regions, as shown in Fig. 7(c). In contrast, we collect all discernible keypoints in the keypoint detection process, thus addressing the problem of insufficient keypoints. Therefore, even for the highly smooth tampered regions, there are still enough discernible keypoints, as shown in Fig. 7(d). Next, because of adopting NMS algorithm and OpponentSIFT descriptor, we acquire uniformly distributed keypoints, and achieve a good matching performance, as shown in Fig. 7(e). In this way, we achieve a good detection performance, as shown in Fig. 7(f).

For the undetected image, although there are a number of keypoints in the tampered regions, the number of matched pairs is too small to detect the tampered regions, as shown in Fig. 8.

Next, for the 10 tampered images involving similar but genuine regions, the proposed method shows a poor detection performance, as seen in Table 3. Through the analysis of the detection process, we find that the detection performance is affected by similar but genuine regions in the feature matching and region correlation map stages, as shown in Fig. 9. Although the clustering of the matched pairs can effectively reduce the influence

of similar but genuine regions (e.g., Fig. 9(c)), there is no effective solution at present for the region correlation map stage. This is a challenge in the keypoint-based methods [30], and will be the focus of future work.

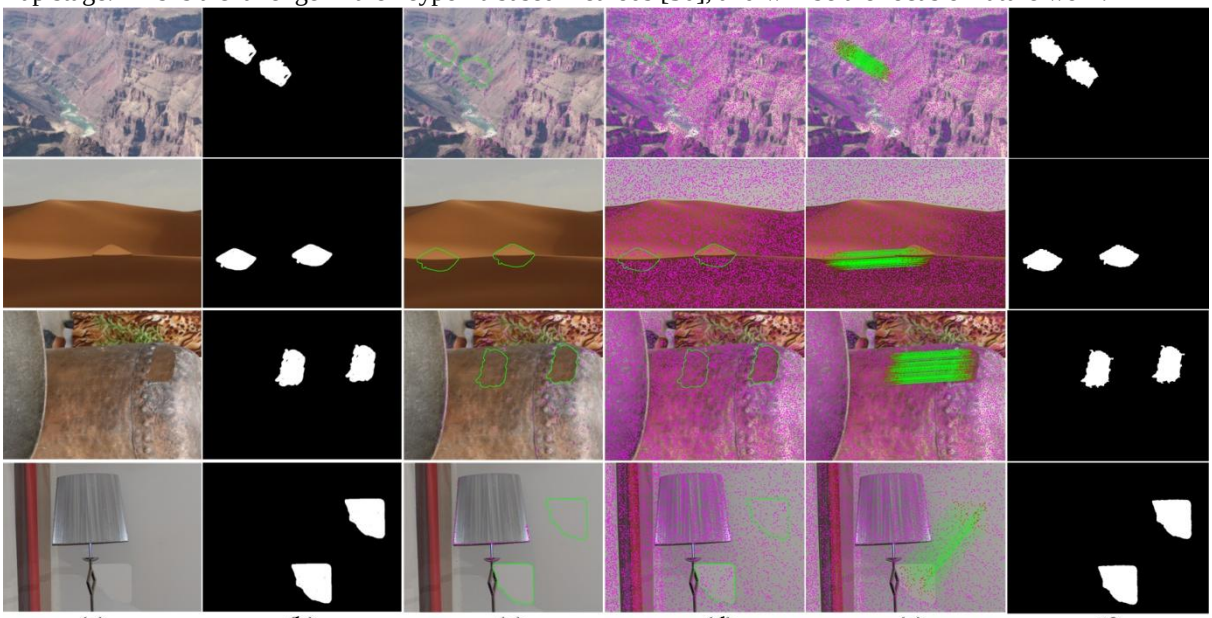


Fig. 7. Detection results in smooth tampered regions: (a) test image, (b) ground truth, (c) keypoints detected by the JLC method, (d) discernible keypoints detected by our proposed method, (e) selected keypoints and matched pairs, (f) detection result.

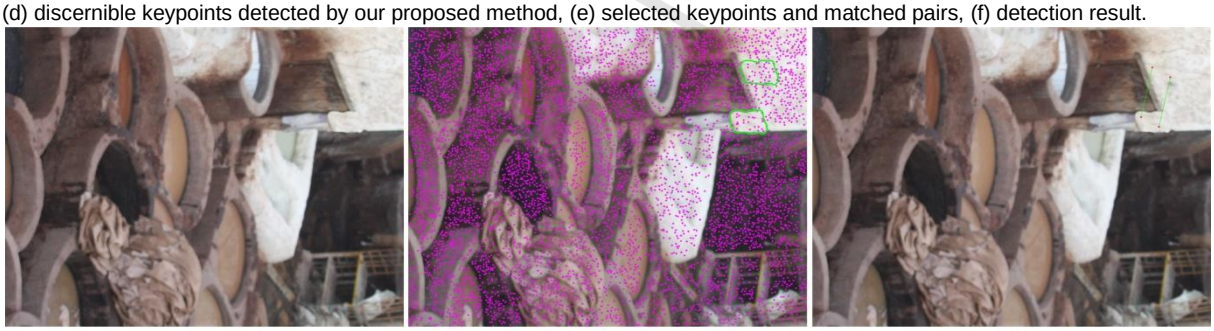


Fig. 8. Example of undetected image: (a) test image, (b) detected SIFT keypoints, (c) matched pairs.

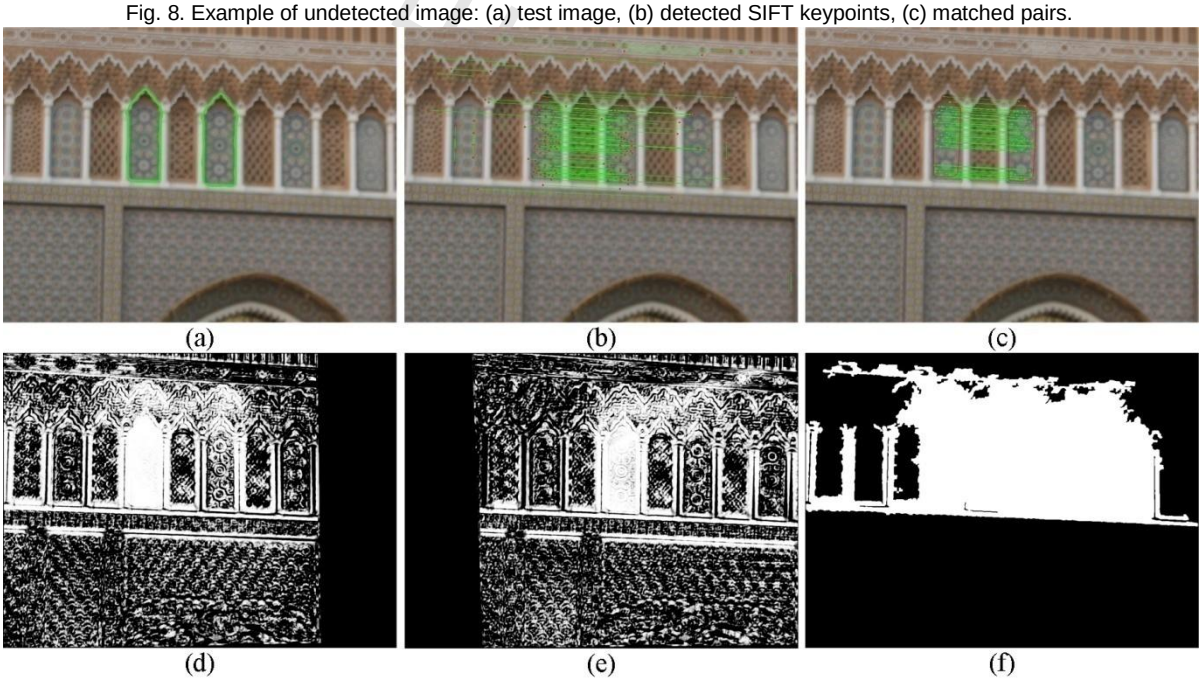
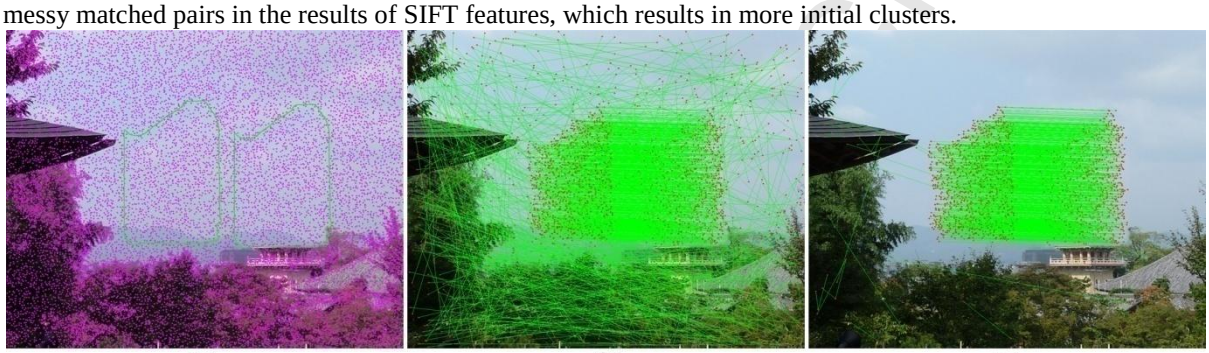


Fig. 9. Effect by similar but genuine regions: (a) test image and tampered regions, (b) matched pairs, (c) clusters after clustering, (d) region correlation map by matrix H, (e) region correlation map by matrix H^{-1}, (f) detection result.

4.4.2. Effect of OpponentSIFT descriptor

In this test, we further study the effect of the OpponentSIFT descriptor by adopting the NMS algorithm and the SIFT descriptor. As mentioned above, the OpponentSIFT descriptor has superior discriminative power and can achieve better matching performance, which has two effects in our method. One direct effect is to improve the detection performance, as shown in Table 3. Compared with the SIFT descriptor, the detection performance achieved with the OpponentSIFT descriptor is superior to some extent. For example, the precision increases by 3.4%, and the recall increases by 1.9%.

Another effect is to reduce the number of initial clusters obtained by the matched pair grouping algorithm. Fig. 10 shows the feature matching results, and Table 4 presents the comparison results with respect to the matched pairs and initial clusters. Compared with the results of the OpponentSIFT descriptor, there are many messy matched pairs in the results of SIFT features, which results in more initial clusters.



(a) (b) (c)

Fig. 10. Comparison matching results between SIFT and OpponentSIFT: (a) keypoints selected by NMS, (b) matched results of SIFT descriptor, (c) matched results of OpponentSIFT descriptor.

Table 4 Comparison results between SIFT and OpponentSIFT descriptors

Descriptor type	Number of matched pairs	Number of initial clusters
SIFT	2987	850
OpponentSIFT	2204	119

4.5. Evaluation of optimized J-Linkage algorithm on FAU

On the dataset FAU, we further evaluate the effectiveness of the measures applied to improve the classical implementation framework, especially the optimized J-Linkage algorithm. Because the AHC and JLC methods are the representative methods chosen from the numerous keypoint-based methods using clustering, we compare the proposed method with the AHC and JLC methods in terms of the clustering time and detection performance.

4.5.1. Test of clustering time

For the comparison of clustering time, we select 10 tampered images from the dataset FAU according to the number of matched pairs. Table 5 shows the results of the clustering times. Our experimental platform is a computer with an Intel core I7-6500 at 2.50 GHz CPU with 8-GB RAM.

Table 5 Clustering times of three methods

Image size	Matches	AHC(s)	JLC(s)	Proposed(s)		
				Segmentation	Clustering	Total
3888×2592	986	5.04	25.38	8.18	0.15	8.33
3264×2448	1576	19.59	81.82	7.03	0.71	7.74
2613×3900	2186	53.75	227.46	9.92	1.24	11.16
3039×2014	2778	115.37	491.14	5.02	0.92	5.94
3039×2014	3366	205.51	775.02	5.15	1.15	6.30
3264×2448	3952	321.56	1189.71	6.47	8.42	14.89
3039×2014	4704	576.12	2114.06	5.07	2.27	7.34
3264×2448	6036	1223.29	4691.67	6.74	4.51	11.25
3264×2448	7584	2487.92	9532.56	6.36	6.02	12.38

3039×2014	8466	3454.64	11880.60	5.03	8.14	13.17

The test results confirm that the clustering times of the AHC and JLC methods increase significantly as the number of matched pairs increases, which agrees with our previous theoretical analysis. The time required for our proposed method consists of the segmentation time and clustering time. The segmentation time is relatively fixed, and the clustering time depends on the number of initial clusters. As described in Section 4.4.2, we obtain a good matching performance by applying the OpponentSIFT descriptor, so that the number of initial clusters is small and the required clustering time is short.

4.5.2. Test of detection performance

In this subsection, the detection performances of the keypoint-based methods for rotation and scaling operations are tested.

For the 384 images with a rotation of the cloned region (orientations of 2^0, 4^0, 6^0, 8^0, 10^0, 20^0, 60^0, and 150^0), the test results are presented in Fig. 11. According to the experimental results, the SIFT-based methods are fairly robust to all the diverse rotations, and our proposed method achieves the best detection performance. However, when the rotation angle is relatively large, such as 60^0, some images in which the tampered regions overlap seriously affect the recall.

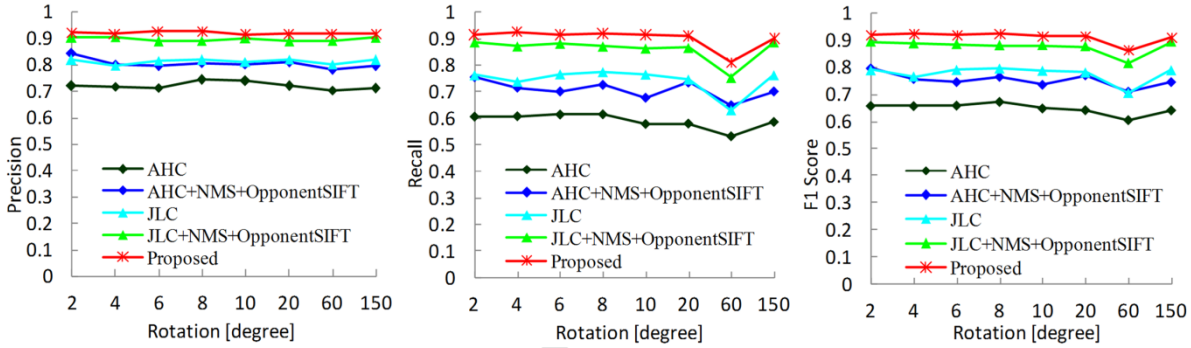


Fig. 11. Comparison results of the rotation operations

For the 672 images with a resizing of the cloned region (scaling factors of 50%, 80%, 91%, 93%, 95%, 97%, 99%, 101%, 103%, 105%,107%, 109%, 120%, and 200%), the test results are presented in Fig. 12. The experimental results show that the performance of the SIFT-based methods remains relatively stable across all scaling parameters, and our proposed method achieves the best detection performance. Moreover, the zoom-out operations have a greater impact on the detection performance than the zoom-in operations, such as scaling to 50%, because the small tampered region becomes even smaller, which affects the detection performance.

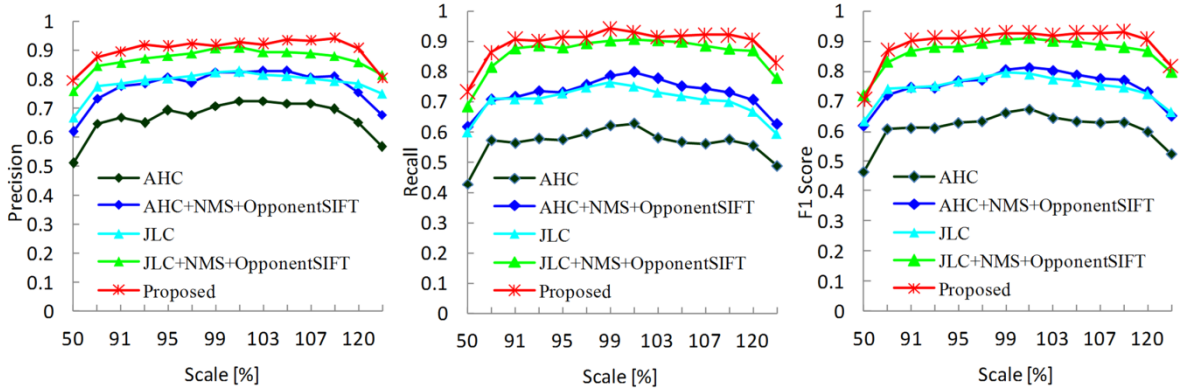


Fig. 12. Comparison results of the scaling operations

These two experimental results show that our proposed method outperforms the AHC and JLC methods in terms of the detection performance, and adopting the NMS selection strategy and OpponentSIFT descriptor can also significantly improve the detection performances of the AHC and JLC methods. However, it should be noted that the number of transformation models of the J-Linkage clustering in the JLC method is set to 500, which is the parameter value in [6]. If the number of transformation models increases further, then the detection

performance of the JLC method will increase step by step, and finally achieve a detection performance to match our proposed method.

To summarize, the experimental results demonstrate that the optimized J-Linkage algorithm can effectively reduce the computing time required for clustering, whilst preserving the detection performance.

4.6. Comparison of detection results on MICC-600

The previous experiments have demonstrated that our measures for improving the classical implementation framework are highly effective in terms of the detection performance and computing time. Therefore, in this test, we compare the proposed method with the SILVA and SegmentBased methods on the dataset MICC-F600. These two methods also apply keypoint detection and feature matching techniques, but use different methods to localize the duplicated regions. The test results are presented in Table 6. Among them, the detection results of the SegmentBased method are taken from [22]. Moreover, the average time required for the proposed method is 111.36S, and the average time of SILVA is 513.81S.

Table 6 Comparison results with SILVA and SegmentBased methods

Method	Precision	Recall	F_1
Proposed	0.902	0.937	0.919
SILVA	0.719	0.740	0.729
SegmentBased	0.860	0.880	0.870

According to the test results, our proposed method has advantages in terms of both the detection performance and computing time. Moreover, by analyzing the test data, we find that there are two factors that affect the detection performance. First, for tampered images involving similar but genuine regions, such as the images named "knight_moves" and "statue" in the dataset MICC-F600, the achieved precisions are only 0.065 and 0.085, respectively. Second, for multiple copies tampered images, the average precision is only 0.835, which should be further improved. These challenges will be the focus of our future work.

5. Conclusions

This paper presents an improved method for SIFT-based copy-move forgery detection. The main contributions of this paper can be summarized in terms of the following three aspects. First, we analyze the failure cause of keypoint-based methods in smooth tampered regions, which is not the insufficient keypoints, but rather an unsuitable method of selecting keypoints. Second, we propose a new solution for obtaining uniformly distributed keypoints in the test image, and adopt the OpponentSIFT descriptor to enhance the discriminative power of SIFT keypoints, which can significantly improve the detection performance. Third, we optimize the J-Linkage algorithm by means of image segmentation and the matched pair grouping algorithm, which can considerably reduce the clustering time in the case of a mass of matched pairs. Our experimental results confirm that the proposed method outperforms other similar state-of-the-art methods in terms of both the detection performance and computing time.

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] B. Mahdian, S. Saic, A bibliography on blind methods for identifying image forgery, Signal Processing: Image Communication 25 (6) (2010) 389-399.

[2] G.K. Birajdar, V.H. Mankar, Digital image forgery detection using passive techniques: A survey, Digital Investigation 10 (3) (2013) 226-245.

[3] M.A. Qureshi, M. Deriche, A bibliography of pixel-based blind image forgery detection techniques, Signal Processing: Image Communication 39 (2015) 46-74.

[4] V. Christlein, C. Riess, J. Jordan, C. Riess, E. Angelopoulou, An evaluation of popular copy-move forgery detection approaches, IEEE Transactions on information forensics and security 7 (6) (2012) 1841-1854.

[5] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, G. Serra, A SIFT-based forensic method for copy move attack detection and transformation recovery, IEEE Transactions on Information Forensics and Security 6 (3) (2011) 1099-1110.

[6] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, L. Del Tongo, G. Serra, Copy-move forgery detection and localization by means of robust clustering with J-Linkage, Signal Processing: Image Communication 28 (6) (2013) 659-669.

[7] R. Toldo, A. Fusiello, Robust multiple structures estimation with J-Linkage, in: European conference on computer vision, Springer, Berlin, Heidelberg, 2008, pp.537-547.

[8] A.J. Fridrich, B.D. Soukal, A.J. Lukáš, Detection of copy-move forgery in digital images, in: Digital Forensic Research Workshop (DFRWS), Citeseer, 2003.

[9] J. Wang, G. Liu, H. Li, Y. Dai, Z. Wang, Detection of image region duplication forgery using model with circle block, in: International Conference on Multimedia Information Networking and Security (MINES), vol.1, IEEE, Hubei, China, 2009, pp.25-29.

[10] S. Bayram, H.T. Sencar, N. Memon, An efficient and robust method for detecting copy-move forgery, in: International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, Taipei, Taiwan, 2009, pp.1053-1056.

[11] S.-J. Ryu, M.-J. Lee, H.-K. Lee, Detection of copy-rotate-move forgery using Zernike moments, in: Information Hiding, Springer, Berlin, Heidelberg, 2010, pp.51-65.

[12] D. Cozzolino, G. Poggi, L. Verdoliva, Copy-Move Forgery Detection Based On Patchmatch, in: International Conference on Image Processing (ICIP), IEEE, Paris, France, 2014, pp.5312-5316.

[13] H. Huang, W. Guo, Y. Zhang, Detection of copy-move forgery in digital images using SIFT algorithm, in: Pacific-Asia Workshop on Computational Intelligence and Industrial Application (PACIIA),vol.2, IEEE, Wuhan, China, 2008, pp.272-276.

[14] X. Pan, S. Lyu, Region duplication detection using image feature matching, IEEE Transactions on Information Forensics and Security 5 (4) (2010) 857-867.

[15] B.L. Shivakumar, S.S. Baboo, Detection of region duplication forgery in digital images using SURF, International Journal of Computer Science Issues 8 (4) (2011) 199-205.

[16] B.L. Shivakumar, S.S. Baboo, Automated forensic method for copy-move forgery detection based on Harris interest points and SIFT descriptors, International Journal of Computer Applications 27 (3) (2011) 9-17.

[17] J. Gong, J. Guo, Image copy-move forgery detection using SURF in opponent color space, Transactions of Tianjin University, 22 (2016) 151-157.

[18] J.M. Guo, Y.F. Liu, Z.J. Wu, Duplication forgery detection using improved DAISY descriptor, Expert Systems with Applications 40 (2) (2013) 707-714.

[19] L. Yu, Q. Han, X. Niu, Feature point-based copy-move forgery detection: covering the non-textured areas, Multimedia Tools and Applications 75 (2) (2016) 1159-1176.

[20] E. Silva, T. Carvalho, A. Ferreira, A. Rocha, Going deeper into copy-move forgery detection: Exploring image telltales via multi-scale analysis and voting processes, Journal of Visual Communication and Image Representation 29 (2015) 16-32.

[21] C.M. Pun, X.C. Yuan, X.L. Bi, Image forgery detection using adaptive oversegmentation and feature point matching, IEEE Transactions on Information Forensics and Security 10 (8) (2015) 1705-1716.

[22] J. Li, X. Li, B. Yang, X. Sun, Segmentation-based image copy-move forgery detection scheme, IEEE Transactions on Information Forensics and Security 10 (3) (2015) 507-518.

[23] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int.'l Journal of Computer Vision 60 (2) (2004) 91-110.

[24] J. Canny, A computational approach to edge detection, IEEE Transactions on pattern analysis and machine intelligence 6 (1986) 679-698.

[25] K.E.a. van de Sande, T. Gevers, C.G.M. Snoek, Evaluating color descriptors for object and scene recognition, IEEE transactions on pattern analysis and machine intelligence 32(9) (2010) 1582-1596.

[26] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk, SLIC superpixels compared to state-of-the-art superpixel methods, IEEE transactions on pattern analysis and machine intelligence 34 (11) (2012) 2274-2282.

[27] V. Christlein, C. Riess, E. Angelopoulou, On rotation invariance in copy-move forgery detection, in: 2010 IEEE International Workshop on Information Forensics and Security, Seattle, WA, 2010, pp.129-134.

[28] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24(6) (1981) 381-395.

[29] R. C. Gonzalez, R. E. Woods, Digital Image Processing, third ed., Prentice Hall, 2008.

[30] B. Wen, Y. Zhu, R. Subramanian, T.T. Ng, X. Shen, S. Winkler, Coverage-a novel database for copy-move forgery detection, ICASSP (submitted).

*Highlights (for review)

- A method to improve the detection performance in smooth tampered regions is proposed.
- A new solution to acquire uniformly distributed keypoints in a test image is proposed.
- Discriminative power of SIFT keypoints is enhanced using OpponentSIFT descriptors.
- A matched pairs grouping algorithm utilizing the correspondence between superpixels is proposed.
- J-Linkage clustering is optimized by using image segmentation and the matched pairs grouping algorithm.