ICIC Express Letters
Part B: Applications
Volume **8**, Number **10**, October **2017**

ICIC International ©2017 ISSN 2185-2766
pp. **1429–1438**

COPY-MOVE FORGERY DETECTION BASED ON LATCH AND REGION-LIKE GROWING

Xuehua Zhou, Xuanjing Shen, Haipeng Chen* and Daqi Tan

Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education
Jilin University
No. 2699, Qianjin Street, Chaoyang Dist., Changchun 130012, P. R. China
xhzhou15@mails.jlu.edu.cn; *Corresponding author: chenhp@jlu.edu.cn

Received May 2017; accepted July 2017

Abstract. *A novel copy-move forgery detection method using the new feature descriptor-LATCH is proposed in this paper, which can solve the problems of high computation complexity, low accuracy and inaccurate tampered region location. First, keypoints are extracted by the classical SIFT. Then, LATCH features are described for corresponding keypoints and match the LATCH features using the Hamming distance. Subsequently, remove false matching by K-means clustering and estimation of geometric transformation parameters. Finally, in order to locate tampered region accurately, a new recursive method based on region-like growing is proposed. Experimental results show that the proposed method not only is effective for geometric transformation and robust to post-processing, but also has higher accuracy on tampered region location and less time consumption. Besides, it has great performance on the type of hiding object forgery.*
Keywords: Image blind identification, Copy-move forgery, LATCH feature, Clustering, Region-like growing

1. **Introduction.** With the development of image editing software, image forgery has been increasingly easy to perform. Forged images can distort truth and affect judicial impartiality. Therefore, image blind identification has become a hot research field recently. Copy-move forgery detection is one of the most important and popular digital forensic techniques [1]. Numerous methods have been proposed [2], which could be classified into two groups [1]: overlapping blocks and keypoint extraction. However, block-based methods are not robust to geometric transformation and these methods have high computation complexity. Keypoint-based methods were proposed, which made up for the deficiency of block-based methods. In particular, SIFT (scale invariant feature transform) [3] and SURF (speeded up robust features) [4] were widely used among them. The algorithm reached a compromise between block-based methods and keypoint-based methods by applying Delaunay Triangulation [5]. However, the disadvantage of these is that the computation complexity is high and these are ineffective on hidden tampering.

In recent years, binary features are becoming popular owing to rapid extraction and high matching rate [6], such as BRIEF (binary robust independent elementary features) [7], and ORB (oriented FAST and rotated BRIEF) [8]. Zhu et al. proposed a scaled ORB method to make ORB robust against scale attacks [9]. However, the existing binary descriptors are mainly based on comparisons of random pixels. Changing either of the pixels can easily lead to changes in descriptor, thereby reducing its performance. Therefore, drawbacks of these descriptors are that they are sensitive to noise and local appearance variations.

Levi and Hassner proposed LATCH to overcome problems that traditional descriptors have strong sensitivity to noise and local appearance variations, high computation complexity and weak robustness [10]. LATCH is formed by comparing patch triplets to increase robustness to noise and local appearance variations. Moreover, LATCH is a fast and compact binary descriptor that performs better than other pure binary descriptors.

Therefore, LATCH feature can be regarded as alternatives to other descriptors, such as SIFT, SURF and ORB.

In order to avoid the limitations of existing methods and locate tampered region accurately, a novel method based on the new LATCH feature descriptor and region-like growing is proposed in this paper. The region-like growing which replaces traditional region growing, avoids the phenomenon of holes and over segmentation in tampered region location. In addition, gray level and geometric transformation parameters are considered, so the proposed method can locate tampered region more accurately. It is worth noting that the method has good performance on geometric transformation and post-processing. In addition, the new method not only has less time consumption, but also has good performance on the type of hiding object forgery. In total, the proposed forgery detection method performs well in case of simple forgery and complex forgery compared to other methods.

2. **The Proposed Method.** The rest of the paper is structured as shown in Figure 1: keypoint detection (Section 2.1), LATCH feature extraction (Section 2.2), feature matching (Section 2.3), removing false matching (Section 2.4), tampered region location (Section 2.5).

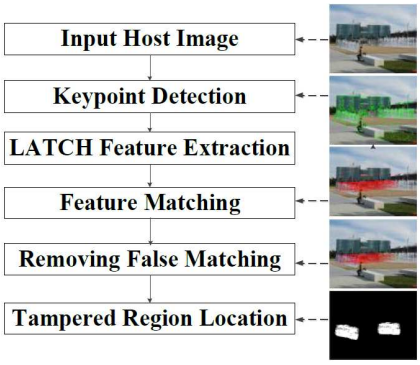


FIGURE 1. Framework of the proposed copy-move forgery detection method

2.1. **Keypoint detection.** There are many methods to detect keypoints such as SIFT [3], SURF [4], and Harris [11]. An analysis of descriptors is provided in [12], which indicates that SIFT is invariant to geometric transformations and robust to post-processing. Owing to good performance and low computation complexity, it has been widely used for image retrieval and object recognition. Therefore, SIFT is used to extract keypoints and a set of keypoints $X = \{x_1, \ldots, x_n\}$ is detected.

2.2. **LATCH feature extraction.** T pixel pairs are considered in traditional binary descriptors. Different descriptors have different selection methods for pixel pairs. Each feature depends on two specific pixels in them. Furthermore, changing any pixel will have a major effect on the feature. So these methods are susceptible to noise and local appearance variations. To avoid the shortcomings, LATCH feature is proposed by Levi and Hassner [10]. It is based on patch triplets, which can provide more information for each comparison. In addition, LATCH is higher robust to Gaussian white noise, Gaussian blur and JPEG compression, and it owns better stability compared with existing descriptors. Therefore, LATCH is an improved binary descriptor which is used to describe keypoints in this part.

Assume $\left\{\widehat{S}_t\right\}_{t=1\ldots T} = \{[P_{t,a}, P_{t,1}, P_{t,2}]\}_{t=1\ldots T}$. $P_{t,a}$, $P_{t,1}$, $P_{t,2}$ are three patches of size 7*7 pixels. $P_{t,a}$ is the central block and $P_{t,1}$, $P_{t,2}$ are its companion patches. Selecting an optimal arrangement from many possible triplet arrangements is important. First, some

arrangements are formed by random selection of $P_{t,a}$, $P_{t,1}$ and $P_{t,2}$. Then evaluate each of these arrangements. Define the quality of an arrangement by summing the number of times it correctly yielded the same binary value. An arrangement is selected if its absolute correlation with all previously selected arrangements is smaller than a threshold ψ which is set to 0.2. Finally, optimal 256 arrangements are selected. The similarities between central patch and two companion patches are estimated by computing the Frobenious norm. Thus, a single binary bit is generated according to Formula (1) as follows.

$$g\left(W, \widehat{S}_t\right) = \begin{cases} 1 & \| P_{t,a} - P_{t,1} \|_F^2 > \| P_{t,a} - P_{t,2} \|_F^2 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where W is a detection window with a predefined size centered on a keypoint, and g is a function which is related to W and $\widehat{S}_t$. Therefore, descriptors $\{f_1, f_2, \ldots, f_n\}$ are extracted, where n is the number of keypoints.

2.3. **Feature matching.** A matching operation is performed among the set of descriptors to identify similar keypoints. First, each descriptor is compared with the rest. Define similarity between two descriptors by Hamming distance. The ith and jth descriptors f_i, f_j are compared by the given Formula (2).

$$Hamming\_distance(x_i, x_j) = \sum_{m=1}^{256} XOR\left(f_i^m, f_j^m\right) \tag{2}$$

where $XOR(a,b) = \begin{cases} 1 & a \neq b \\ 0 & a = b \end{cases}$ and f_i^m is the mth element of the ith descriptor. Then, keypoints (x_i, x_j) are matched if the Hamming distance is less than a threshold ε and save matched pairs as *match*.

2.4. **Removing false matching.** When matched pairs are found, false matching needs to be removed. Firstly, K-means clustering is performed on *match* to locate tampered region initially. Secondly, geometric transformation parameters are estimated by core points. Then, false matched pairs are removed by clustering results and geometric transformation parameters. Figure 2 shows the framework of removing false matching.

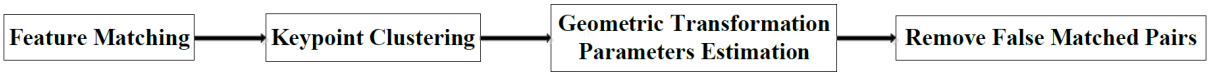


FIGURE 2. Framework of removing false matching

2.4.1. *Keypoint clustering.* To locate tampered region initially, hierarchical agglomerative clustering is used [13]. This method only considers coordinates of matched pairs, while ignoring matching constraint between keypoints. Worse still, computation complexity is high. Therefore, K-means clustering with less time consumption and great clustering performance [14] is applied to matched pairs *match*, and two clusters C and M are obtained.

2.4.2. *Geometric transformation parameters estimation.* Assume that core points C_{core} and M_{core} are the most similar matching points in C and M, as defined in (3).

$$\begin{aligned} C_{core}(x,y) &= \min\left(dis\left(C_i, M_j\right)\right) \quad C_i \in Match; \\ M_{core}(x,y) &= \min\left(dis\left(C_i, M_j\right)\right) \quad M_j \in Match \end{aligned} \tag{3}$$

where C_{core} represents core points and C_i represents keypoints which satisfy the matching conditions in C. The same goes for the definition of M_{core}, M_j. *dis* means distance between two keypoints and function min() means to get minimum.

On the one hand, the close distance between core point and other keypoints will lead to false matching. Therefore, distance between them is limited as defined in (4), where T_h is 10.

$$\begin{aligned}\sqrt{(C_{core}.x - C_i.x)^2 + (C_{core}.y - C_i.y)^2} \geq T_h;\\ \sqrt{(M_{core}.x - M_i.x)^2 + (M_{core}.y - M_i.y)^2} \geq T_h\end{aligned} \tag{4}$$

On the other hand, false matching does not satisfy geometric transformation relation between tampered regions. Consequently, false matching can be removed by geometric transformation parameters SD and θ. Here difference of scale named SD between tampered regions is calculated by Formula (5):

$$SD = \frac{\sum_{i=1}^{n} \sqrt{(C_{core}.x - C_i.x)^2 + (C_{core}.y - C_i.y)^2}}{\sum_{i=1}^{n} \sqrt{(M_{core}.x - M_i.x)^2 + (M_{core}.y - M_i.y)^2}} \tag{5}$$

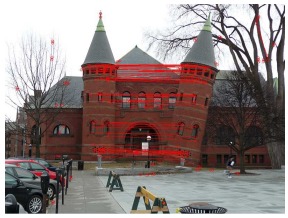
The angle difference θ between tampered regions is calculated by Formula (6):

$$\theta = \frac{\sum_{i=1}^{n} (\alpha_i - \beta_i)}{n} \tag{6}$$

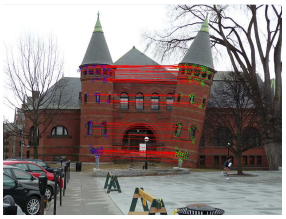
where α is the angle between $line(C_i, E)$ and $line(C_{core}, E)$, β is the angle between $line(M_i, E)$ and $line(M_{core}, E)$, and E is the center point of the image. The effect of removing false matching can be seen in Figure 3(c) and Figure 3(d).

(a) Original image

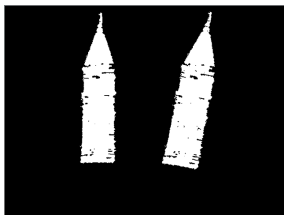
(b) Tampered image



(c) Before removing false matching



(d) After removing false matching



(e) Tampered region

FIGURE 3. The results of removing false matching

2.5. **Tampered region location.** The basic idea of region growing is that the pixels having the same properties such as gray level are merged into one region by examining eight neighborhood pixels around the pixel [15]. However, uneven gray level may lead to holes and over segmentation. In addition, gray level and geometric transformation parameters are indispensable in tampered region location. Therefore, a method of region-like growing is proposed based on region growing, which is another novelty of our work.

The processes of region-like growing are described as follows.

Step 1. Two core points $C_{core}(x, y)$, $M_{core}(x, y)$ are selected as seed pixels.

Step 2. The eight neighborhood pixels $C_i.np$ around core point $C_{core}(x, y)$ are compared with the corresponding eight neighborhood pixels $M_j.np$ around the other $M_{core}(x, y)$ according to a certain rule as defined in Equation (7). The pixel which satisfies the condition is added to the respective regions.

$$Gray(C_i.np) - Gray(M_j.np) < th \tag{7}$$

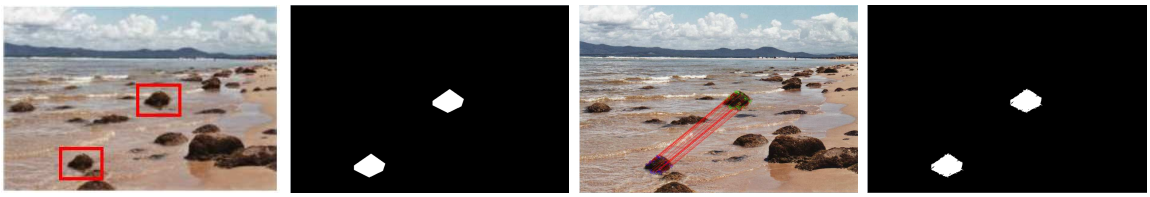
where C_i denotes the ith pixel in C, $C_i.np.x$ satisfies Equation (8), and $C_i.np.y$ satisfies Equation (9). The same goes for the definition of M_i, $M_i.np.x$ and $M_i.np.y$. In addition, $Gray(k)$ denotes the gray level of a certain keypoint k and th is a threshold which is set to 1.

$$M_i.np.x = ((C_i.np.x - C_{core}.x) \times \cos\theta - (C_i.np.y - C_{core}.y) \times \sin\theta) \times S + M_{core}.x \tag{8}$$

$$M_i.np.y = ((C_i.np.x - C_{core}.x) \times \sin\theta + (C_i.np.y - C_{core}.y) \times \cos\theta) \times S + M_{core}.y \tag{9}$$

Step 3. Repeat the above process until the keypoints in Clusters C and M are all visited.

Finally, as can be seen from Figure 4, the result of tampered region location Figure 4(d) is very accurate compared with the ground truth in Figure 4(b).



(a) Tampered image (b) The ground truth (c) After removing false matching (d) Tampered region

FIGURE 4. The results of tampered region location

3. **Experimentation and Evaluation.** To assess effectiveness and robustness, the proposed algorithm is compared with scaled ORB [9] and triangle-based methods [5] on datasets: Dataset [5] and Manipulate [1]. Hardware environment is a notebook computer with Intel Core i3 2.20 GHz processor and software used is Visual Studio 2015 + Opencv 3.1.

3.1. **Evaluation metrics.** The performance of the proposed method is measured with true positive rate (TPR) and false positive rate (FPR), where TPR is the fraction of tampered images correctly identified as such, while FPR is the fraction of original images that are not correctly identified.

3.2. **Experiment and analysis.** The performance of the proposed method is evaluated from four aspects: geometric transformation (Section 3.2.1); post-processing (Section 3.2.2); a class tampering on hiding trace (Section 3.2.3); time consumption (Section 3.2.4).

3.2.1. *Geometric transformation.*

(1) **Naive copy-move.** Basically, evaluate the proposed method under ideal conditions; namely, use 28 original images and 28 naive copy-move images. One of the results is shown in Figures 5(a)-5(e), where red rectangular area designates the copied and pasted regions. The ROC curve is given in Figure 9(a).

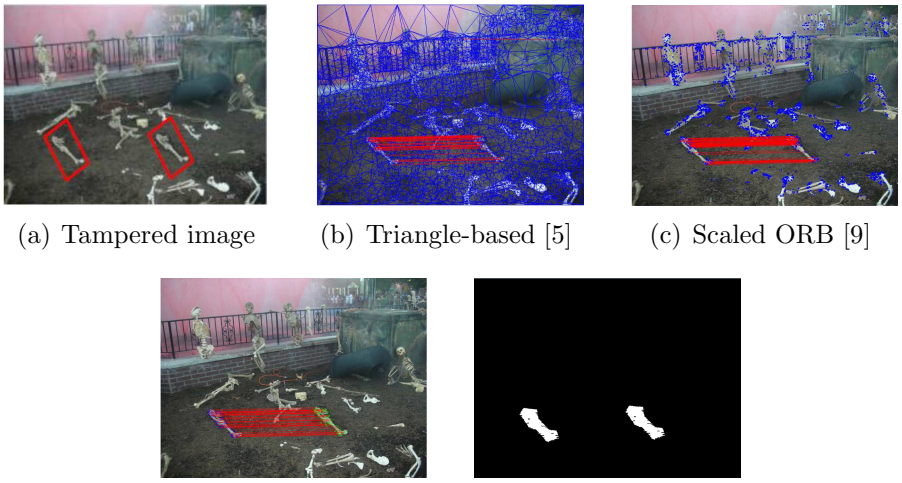
(2) **Scaled copy-move.** Copied regions are scaled with the scale factor varying from 95% to 115%, in steps of 10%, based on 34 images. In this case, a total of $34 * 3 = 102$ images are tested. Figures 6(a)-6(e) show the result of a tampered image whose scaling factor is 0.95. The ROC curves of images which are scaled by 95% and 105% are given in Figure 9(b).

(3) **Rotated copy-move.** 34 tampered images are used by rotating at a randomly chosen angel from the set $\{90°, 180°, 270°\}$. A total of $34 * 3 = 102$ images are tested. Results of different algorithms for image whose rotation angle is 270° are shown in Figures 7(a)-7(e). The ROC curves of images which are rotated by 90° and 180° are given in Figure 9(c).

(4) **Scaled combined with rotated copy-move.** It is not simple to detect combinational geometric transformation. Scaling combined with rotation is applied to 28 images to evaluating the proposed algorithm. In this case, a total of $28*2 = 56$ images are tested. The results of such forgery are shown in Figures 8(a)-8(e).

Comparison between the proposed method and others has been provided in Figures 5-9. *TPR* values indicate that the proposed method has higher accuracy. Furthermore, *FPR* for the proposed method is smaller than triangle-based methods [5] and the method of scaled ORB [9] at the same time. That is to say, the proposed method exhibits better classifying performance compared to other methods for various geometric transformation.

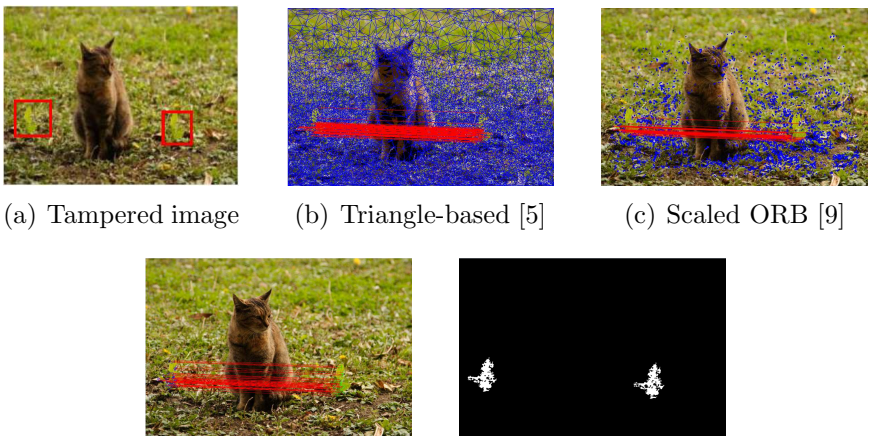
3.2.2. *Post-processing.* The number of images used in Section 3.2.1 is 288. These images were carried out three kinds of post-processing to evaluate robustness. Two parameters are selected, so the total number of test images is $288 * 2 = 576$ for each operation.



(a) Tampered image (b) Triangle-based [5] (c) Scaled ORB [9]

(d) Proposed (e) Tampered region

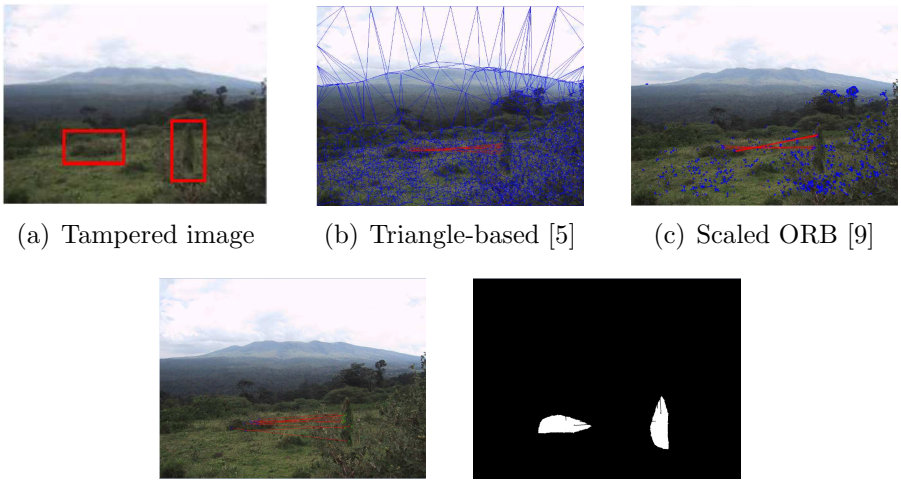
FIGURE 5. The results of naive copy-move



(a) Tampered image (b) Triangle-based [5] (c) Scaled ORB [9]

(d) Proposed (e) Tampered region

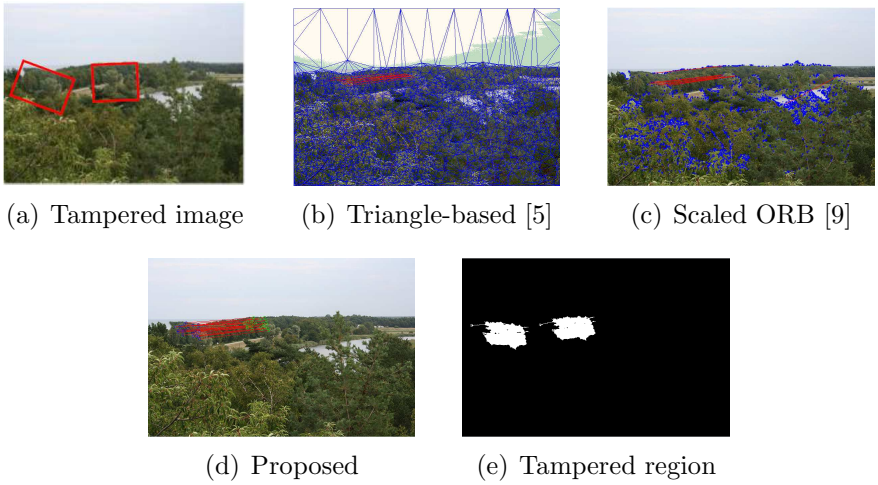
FIGURE 6. The results of image scaled by 95%



(a) Tampered image (b) Triangle-based [5] (c) Scaled ORB [9]

(d) Proposed (e) Tampered region

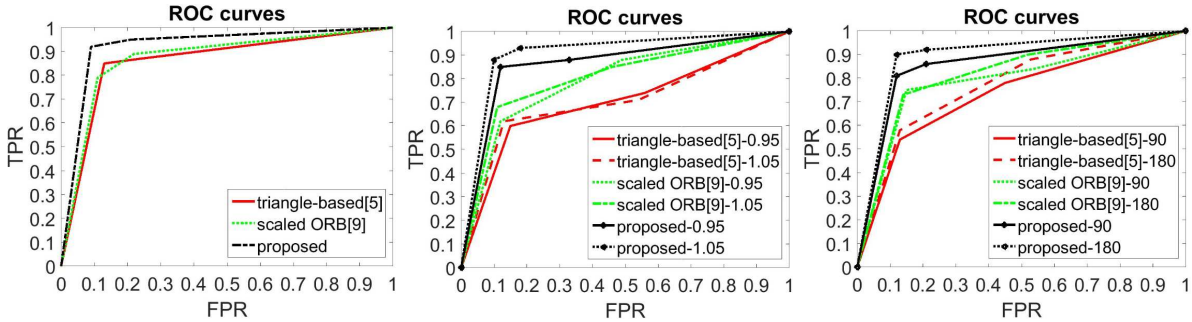
FIGURE 7. The results of image rotated by 270°



(a) Tampered image (b) Triangle-based [5] (c) Scaled ORB [9]

(d) Proposed (e) Tampered region

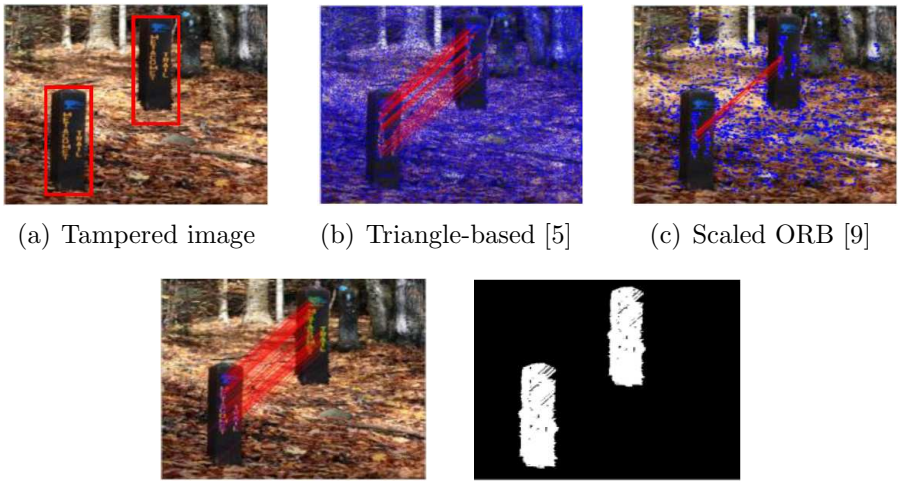
FIGURE 8. The results of image with combinational geometric transformation



(a) Naive copy-move (b) Scaled copy-move (c) Rotated copy-move

FIGURE 9. ROC curves based on triangle-based [5], scaled ORB [9] and proposed method

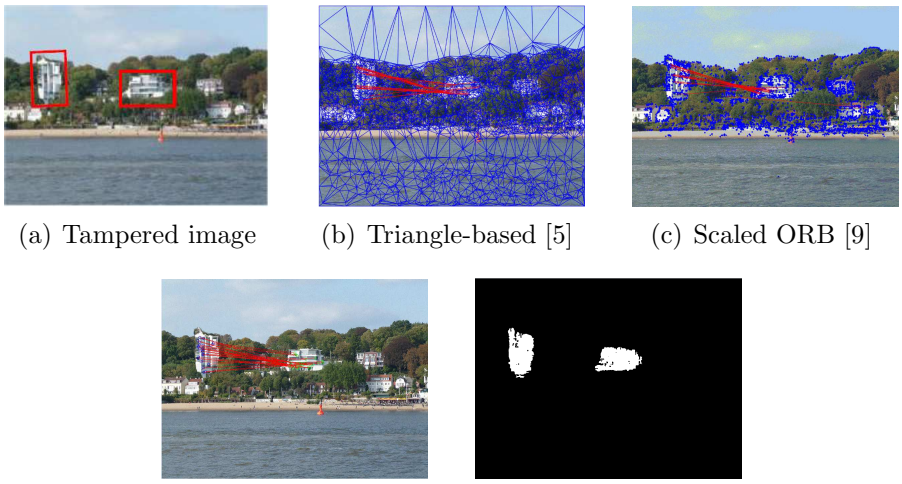
(1) **Gaussian blur.** Forged images are blurred by Gaussian function with window size $w = 3$, $\sigma = 0.5$ and $w = 3$, $\sigma = 2$. Figures 10(a)-10(e) show the result of a forged image in blur-2 and Figure 13(a) shows the ROC curves.



(a) Tampered image (b) Triangle-based [5] (c) Scaled ORB [9]

(d) Proposed (e) Tampered region

FIGURE 10. The results of image in Gaussian blur with window size $w = 3$, $\sigma = 2$



(a) Tampered image (b) Triangle-based [5] (c) Scaled ORB [9]

(d) Proposed (e) Tampered region

FIGURE 11. The results of image in Gaussian white noise with $m = 0$, $v = 0.0005$

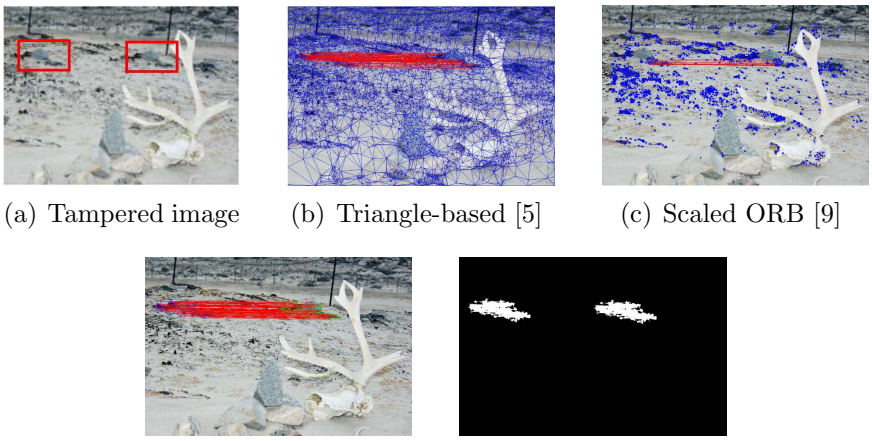
(2) **Gaussian white noise.** Forged images are added Gaussian white noise with mean $m = 0$, $v = 0.001$ and $m = 0$, $v = 0.0005$. The results are shown in Figures 11(a)-11(e) and Figure 13(b).

(3) **JPEG compression.** Forged images are resaved with JPEG quality factor 50 and 80. The result of a forged image in JPEG50 is shown in Figures 12(a)-12(e), and the ROC curves are shown in Figure 13(c).

Forged images with post-processing are experimented. It can be observed that TPR of the proposed method exceeds triangle-based methods [5] and the method of scaled ORB [9] by a large amount, especially for forged images which are added Gaussian white noise. Therefore, the proposed method has great robustness for post-processing.

3.2.3. *Hidden object forgery.* Hidden object forgery aims to hide an object by using a duplication of smooth region. Results are provided in Figure 14 which shows that the proposed method has good adaptability even for hidden object forgery.

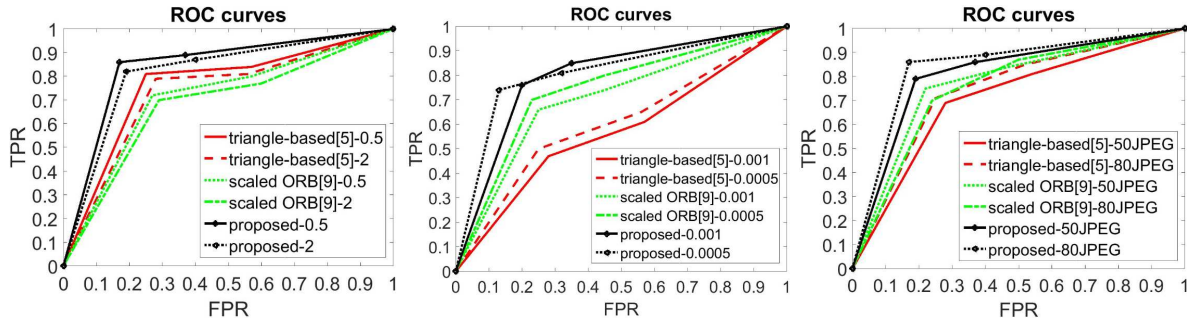
3.2.4. *Time consumption.* Time of extraction and matching is counted respectively thr-ough images in Section 3.2 for comparison and result is shown in Table 1. As can be seen,



(a) Tampered image (b) Triangle-based [5] (c) Scaled ORB [9]

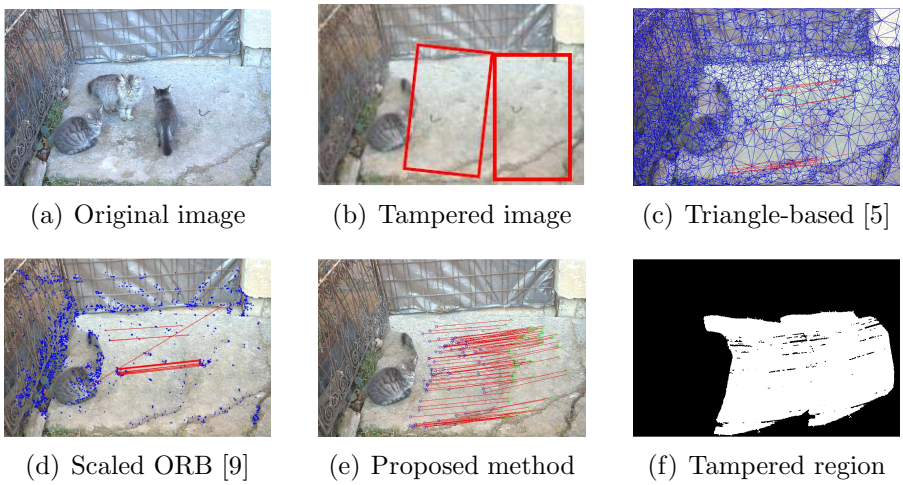
(d) Proposed (e) Tampered region

FIGURE 12. The results of image in JPEG with quality factor 50



(a) Gaussian blur (b) Gaussian white noise (c) JPEG compression

FIGURE 13. ROC curves based on triangle-based [5], scaled ORB [9] and proposed method



(a) Original image (b) Tampered image (c) Triangle-based [5]

(d) Scaled ORB [9] (e) Proposed method (f) Tampered region

FIGURE 14. The result of hidden object forgery

feature extraction time is only slightly longer than binary features and far faster than triangle-based [5]. Time for matching is the shortest. All in all, the proposed method is thirty-six orders of magnitude faster than triangle-based method [5], and twenty times faster than scaled ORB [9].

TABLE 1. The comparison of running time

Methods	Respective running time (average)		Total running time
	Feature extraction	Matching	
Triangle-based [5]	62.28s	127.34s	189.62s
Scaled ORB [9]	0.97s	104.57s	105.54s
Proposed	1.05s	4.18s	5.23s

4. **Conclusions.** A new copy-move forgery detection method based on LATCH and region-like growing has been proposed. The method yields better discriminative capability even if geometric transformations and post-processing are applied. In addition, the presented method shows effectiveness in hiding object forgery when compared with scaled ORB [9] and triangle-based method [5]. Tampered region can be located accurately while ensuring the less time consumption and high accuracy, by applying a recursive region-like growing. Future work will be dedicated to studying comprehensive authentication algorithm for multiple forgery methods rather than just copy-move forgery.

Acknowledgment. This research is supported by the National Natural Science Foundation of China (No. 61672259), the National Youth Science Foundation of China (No. 61602 203), and the Natural Science Foundation of Jilin Province (No. 20150101055JC).

REFERENCES

[1] V. Christlein, C. Riess, J. Jordan et al., An evaluation of popular copy-move forgery detection approaches, *IEEE Trans. Information Forensics & Security*, vol.7, no.6, pp.1841-1854, 2012.

[2] J. A. Redi, W. Taktak and J. L. Dugelay, Digital image forensics: A booklet for beginners, *Multimedia Tools and Applications*, vol.51, no.1, pp.133-162, 2011.

[3] D. G. Lowe, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, vol.60, no.2, pp.91-110, 2004.

[4] H. Bay, T. Tuytelaars and L. V. Gool, SURF: Speeded up robust features, *Computer Vision & Image Understanding*, vol.110, no.3, pp.404-417, 2006.

[5] E. Ardizzone, A. Bruno and G. Mazzola, Copy-move forgery detection by matching triangles of keypoints, *IEEE Trans. Information Forensics & Security*, vol.10, no.10, pp.2084-2094, 2015.

[6] J. Heinly, E. Dunn and J. M. Frahm, Comparative evaluation of binary features, *Computer Vision – ECCV*, pp.759-773, 2012.

[7] M. Calonder, V. Lepetit, V. C. Strecha et al., BRIEF: Binary robust independent elementary features, *European Conference on Computer Vision*, pp.778-792, 2010.

[8] E. Rublee, V. Rabaud, K. Konolige et al., ORB: An efficient alternative to SIFT or SURF, *IEEE International Conference on Computer Vision*, pp.2564-2571, 2012.

[9] Y. Zhu, X. Shen and H. Chen, Copy-move forgery detection based on scaled ORB, *Multimedia Tools and Applications*, vol.75, no.6, pp.3221-3233, 2016.

[10] G. Levi and T. Hassner, LATCH: Learned arrangements of three patch codes, *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp.1-9, 2016.

[11] C. Harris and M. Stephens, A combined corner and edge detector, *Proc. of the 4th Alvey Vision Conference*, vol.15, pp.147-151, 1988.

[12] K. Mikolajczyk and C. Schmid, A performance evaluation of local descriptors, *IEEE Trans. Pattern Analysis & Machine Intelligence*, vol.27, no.10, pp.257-263, 2005.

[13] I. Amerini, L. Ballan, R. Caldelli et al., A SIFT-based forensic method for copy-move attack detection and transformation recovery, *IEEE Trans. Information Forensics & Security*, vol.6, no.3, pp.1099-1110, 2011.

[14] S. L. Yang, Y. S. Li, X. X. Hu et al., Optimization study on k value of K-means algorithm, *Systems Engineering-Theory & Practice*, vol.26, no.2, pp.97-101, 2006.

[15] R. Adams and L. Bischof, Seeded region growing, *IEEE Trans. Pattern Analysis & Machine Intelligence*, vol.16, no.6, pp.641-647, 1994.