EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Copy-move forgery detection for image forensics using the superpixel segmentation and the Helmert transformation

Hui-Yu Huang* and Ai-Jhen Ciou

Abstract

The increasing popularity of the internet suggests that digital multimedia has become easier to transmit and acquire more rapidly. This also means that this multimedia has become more susceptible to tampering through forgery. One type of forgery, known as copy-move duplication, is a specified type that usually involves image tampering. In this study, a keypoint-based image forensics approach based on a superpixel segmentation algorithm and Helmert transformation has been proposed. The purpose of this approach is to detect copy-move forgery images and to obtain forensic information. The procedure of the proposed approach consists of the following phases. First, we extract the keypoints and their descriptors by using a scale-invariant feature transform (SIFT) algorithm. Then, based on the descriptor, matching pairs will be obtained by calculating the similarity between keypoints. Next, we will group these matching pairs based on spatial distance and geometric constraints via Helmert transformation to obtain the coarse forgery regions. Then, we refine these coarse forgery regions and remove mistakes or isolated areas. Finally, the forgery regions can be localized more precisely. Our proposed approach is a more robust solution for scaling, rotation, and compression forgeries. The experimental results obtained from testing different datasets demonstrate that the proposed method can obtain impressive precision/recall rates in comparison to state-of-the-art methods.

Keywords: Image forensics, Copy-move forgery detection, Tampering detection, Region duplication, Superpixels

1 Introduction

As a result of technological advances and the convenience of the internet, human beings are now able to easily access interesting multimedia from the internet and remake or tamper with it as they see fit. Copy-move forgery imaging is a special type of forgery that involves copying parts of an image and then pasting the copied parts into the same image. Hence, image forensics associated with copy-move forgery detection have become increasingly important in our networked society. The technology used in image forensics can be categorized into passive detection or active detection [1]. The active detection method requires prior information derived from an image to identify the image authenticity, such as watermarking. Contrary to active detection methods, passive detection methods are not required to obtain previous information on an image. Passive detection methods can utilize the advantages of the detective strategy to find the tampering regions. Hence, a large majority of image forgery detection methods adopt a passive-based strategy to perform the type of tampering identification discussed in the present study. Passive detection technology can be categorized into block-based methods [2–10] and keypoint-based methods [11–21]. In the present study, we focus on the keypoint-based approach.

Block-based methods segment an image into overlapping blocks and then extract features from those blocks. The forgery regions are determined by computing the similarity between block features. Wang et al. [2] proposed block-based forensics to detect region duplication

* Correspondence: hyhuang@nfu.edu.tw
Department of Computer Science and Information Engineering, National Formosa University, 64, Wun-hua Rd, Huwei, Yunlin 632, Taiwan

for an image. The method mainly used the mean intensities of a circle with different radii around the center of the block to represent the features of the block. Ryu et al. [3, 4] used Zernike moments as block features. The method can identify the forged region by copy-rotate-move forgery. Huang et al. [5] proposed a discrete cosine transform (DCT)-based forgery detection method. The image is first divided into overlapping blocks and the DCT is applied, thus the DCT coefficients for each block are quantized by fixed stepsize q and then rounded to the nearest integer. A row vector as block feature can then be obtained by using a zigzag scan. The duplicated image blocks are compared in the matching step. This method can detect JPEG compression, but the DCT-based feature vector cannot resist geometrical tampering.

Wang et al. [6] proposed a forgery method that combines the discrete wavelet transform (DWT) and the DCT. The DWT and DCT are applied to each image block to extract features. The coefficients obtained by the DWT and DCT are multiplied to form the eigenvectors. Then, the similarity of two blocks is estimated, along with the mean and variance distances between the eigenvalues in their corresponding eigenvectors. This method can resist JPEG compression but not image processing operations.

Bravo-Solorio and Nandi [7] proposed a polar-based forgery detection method to detect copy-move attacks for an image. This method subdivided an image into overlapping blocks of pixels. The pixels within the block are first transformed into log-polar maps (LPM), and then summed along the angle axis, to generate one-dimensional descriptors. Subsequently, they will compute the Fourier coefficient magnitude after Fourier transformation. The descriptors are invariant to reflection and rotation. The descriptor of each block is used to compute the information entropy as block features. By computing the entropy difference between blocks, the similar regions are found. However, a significant amount of smooth duplication regions may arise during mistake detection.

Davarazni et al. [8] used multiresolution local binary patterns (MLBP) for forgery detection. This method used LBP operations to extract feature vectors for each block, and then sorted these vectors based on lexicographical order. The duplicated image blocks are detected in the block matching step using a k-tree. This method is time consuming and does not detect any rotation angles for duplication regions. Lee et al. [9] used a histogram of oriented gradients (HOG) of each block as features; these features are ordered by using lexicographical sorting. The duplicated image blocks are detected by measuring similar block pairs. Li et al. [10] used a polar harmonic transform to extract the rotation and scaling invariant features as block features (similar to the method of Lee et al. [9]). These feature vectors are lexicographically sorted, and the forged regions are detected by finding similar block pairs.

In keypoint-based methods, image features are extracted and matched with the entire image to identify the regions that were tampered with. Common and well-known feature points have scale-invariant feature transform (SIFT) [22] features and speedup robust features (SURF) [23]. These feature points have been widely used for image retrieval and object recognition because of their robustness in geometrical transformations (e.g., scaling and rotation). Based on these advantages, these features have been applied to digital forensics. In [11, 12], these methods applied a SIFT to the host image to extract keypoints, which were then matched to one another. When the value of the displacement vector exceeded the threshold, the sets of corresponding SIFT keypoints are labeled as the tampered regions. The method used for combining the SIFT keypoints and J-linkage algorithm to localize the forgery regions has been reported [13].

In [14, 15], the SURFs were applied to extract the keypoint features, which makes it possible to detect duplicated regions of various sizes. Additionally, Mishra et al. [15] also used hierarchical agglomerative clustering (HAC) to group the matched keypoints from these sets of keypoints. Several different technologies based on SURF and SIFT and Harris were applied in [16, 17]. Pun et al. [18] integrated both the block-based and keypoint-based methods to detect the forged regions. Several keypoint-based methods involved with segmentation methods have been reported in the following references: [19–21, 24–27]. Christlein et al. [28] evaluated the performance of feature sets in existing copy-move forgery detection algorithms.

Many methods from the literature deal only with simple copy-move forgery scenarios, while other approaches present relevant contributions toward the detection of sophisticated tampering. However, these approaches still have major limitations. Most of the current block-based methods use a similar framework; the main differences between frameworks are that they use different feature extraction methods to extract the block features. The block-based detection of forgery regions can be time-consuming because the host image is divided into overlapping blocks, and they cannot detect geometrical transformations of the forgery regions. In contrast, the keypoint-based forgery detection methods can detect geometrical transformations and require less computational resources; however, they do not have good localization power. Thus, there is room for improving true positive rate (TPR) results. Based on the above reasons, we propose conducting image forensics based on an simple linear iterative clustering (SLIC) algorithm [29] and Helmert transformation [30] to achieve copy-move forgeries with rotations, resizings, and combinations of the two. This proposed scheme uses the SIFT

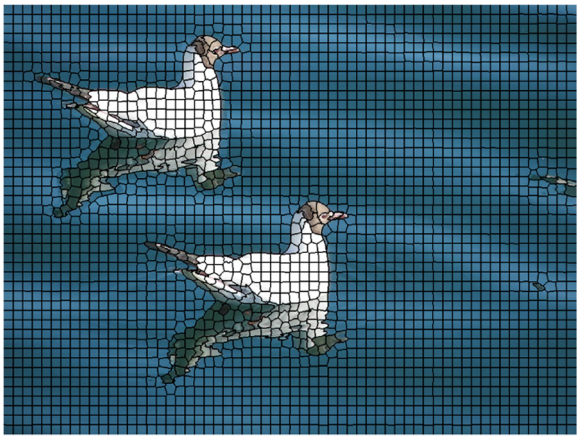

Fig. 1 An example of SLIC segmentation. Every superpixel is approximately the size of 300 pixels

algorithm to extract the keypoints from an image and then designs our algorithm. Our approach can efficiently resist geometrical transformations and JPEG compressions, and localize the forgery regions more precisely at a reasonable computational cost.

The rest of this study is organized as follows. Section 2 presents the related techniques. The proposed method is described in Section 3. In Section 4, we present the experimental results to verify the robustness of the proposed algorithm. Finally, Section 5 concludes this study.

2 Related techniques

In this section, we briefly describe the related methods that apply to our proposed approach.

2.1 Superpixel segmentation

One type of image segmentation method is called the superpixel segmentation method. It groups the pixels of an image into perceptually meaningful atomic regions that can be used to replace the rigid structure of the pixel grid. A simple linear iterative clustering (SLIC)-based superpixel algorithm is proposed by Achanta et al. [29]. It uses a *k*-means clustering approach to efficiently generate superpixels, and it can adhere to the boundaries very well. The only parameter (k) in the SLIC algorithm is to assign the desired number of approximately equally sized superpixels. The algorithm is briefly described in the following paragraph. Details of the procedures have been reported in [29].

This SLIC algorithm is adopted in CIELAB color space. The SLIC algorithm adapts a *k*-means clustering approach to efficiently generate the superpixels, and it adheres to the boundaries very well. First, the clustering procedure begins with an initialization step where the k initial cluster centers, where (l, a, b) are the three color components of a pixel, and (x, y) are its two spatial coordinates, are sampled on a regular grid (called a superpixel), spaced S pixels apart. The S interval is $\sqrt{N/k}$, in which N represents the number of pixels for an image. In order to avoid centering a superpixel on an edge or on a noisy pixel, the centers are moved to seed locations corresponding to the lowest gradient position in a $n \times n$ neighb$C_i = [l_i, a_i, b_i, x_i, y_i]^T$, $i = 1, 2, \cdots, k$,orhood. As is known to us, the edge or noisy pixel is often positioned on a pixel point that has the largest gradient variation. Therefore, selecting the lowest gradient pixel point to position the center for a superpixel can efficiently reduce the chance of seeding a superpixel with an edge or a noisy pixel.

Additionally, in order to speed up the SLIC algorithm, the search area is reduced to the size of $2S \times 2S$ around the superpixel center, in contrast to the traditional *K*-means clustering method. Then, by computing the distance between the center point and other pixel points within the cluster, an update step adjusts the cluster centers to be the mean vector of all the pixels belonging to the cluster, once each pixel has been associated to the nearest cluster center. The residual error is computed by means of the L_2 norm between the new cluster center locations and previous cluster center locations. Finally, the assignment and updated steps can be repeated iteratively until the error converges. As [29] discussed, after

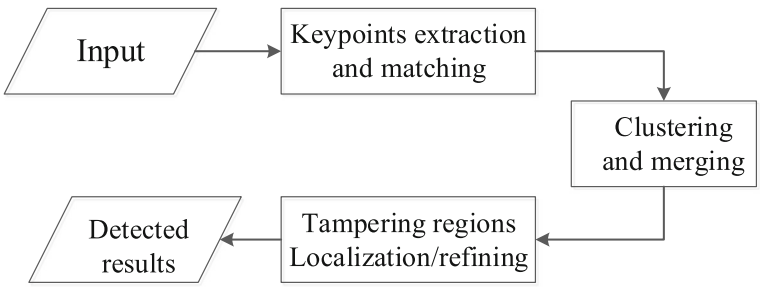

Fig. 2 The flowchart of the proposed system

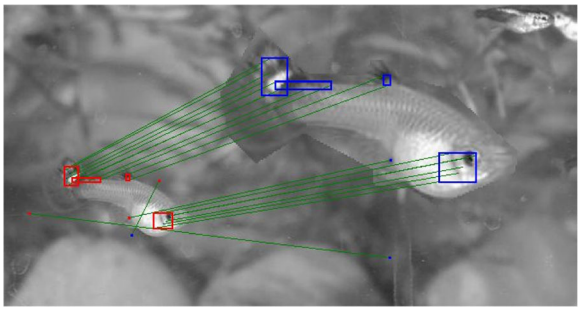

Fig. 3 Clustering result based on [16]. The matching pairs belonging to the same group are segmented into different groups, such as red subgroups or blue subgroups

iterating ten times, most images can achieve the convergence. Figure 1 shows an example of SLIC segmentation for a superpixel that is roughly the size of 300 pixels.

2.2 Helmert transformation

In our work, since all points lie in a plane, the Helmert transform becomes transformations from one rectangular coordinate system to another rectangular system. These transformations include rotation, scaling, and translations for all points. The transformation equations can be formed in matrix notation using mathematical operations [30].

$$\begin{bmatrix} X_p \\ Y_p \end{bmatrix} = \begin{bmatrix} A & B \\ -B & A \end{bmatrix} \begin{bmatrix} x_p \\ y_p \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}, \quad (1)$$

where (x_p, y_p) coordinates are transformed into (X_p, Y_p) coordinates by the addition of translations t_x and t_y. A and B are the transformation parameters. This transformation is called the Helmert transformation [30], also known as similarity transformation. Helmert transformations have a lower degree of freedom, therefore they have lower computational complexity available to transform the coordinates of points in one point (x, y) into coordinates in another point (X, Y). As shown in Eq. (1), only four parameters are needed to compute the coordinate transformations, such as rotation, scaling, and translations. In addition, a well-known transformation known as the affine transformation usually uses map coordinate transformations. However, affine transformations require six parameters to achieve transformations. The advantages of the Helmert transformation include not only resistance to rotation, scaling, and translations, but also reduced computational complexity. For instance, given the coordinates of two pairs, we can obtain four parameters of Helmet transformation by Eq. (1). Hence, in our experiments, we adopt the Helmert transformation instead of affine transformation to acquire the coordinates after transformation.

3 Proposed method

In this study, we propose keypoint-based image forensics based on the Helmert transformation and SLIC algorithm. The main procedures include keypoint extraction and matching, clustering and group merging, and forgery region localization and refining. Figure 2 illustrates the flowchart of the proposed system. Details of procedures are described in the following subsections.

3.1 Keypoint extraction and matching

Based on the SIFT algorithm [22], we can obtain all candidates of keypoints and the corresponding descriptors for an image. Using these candidates, we will search for the best matching pairs to perform additional grouping.

First, each keypoint within all candidates will compute the Euclidean distance between other keypoints via corresponding descriptors, and will also perform the matching operation. The nearest neighbor distance ratio (NNDR) [31], which is the ratio of the smallest distance to the second-smallest distance, is used to perform the matching. This ratio is depicted as

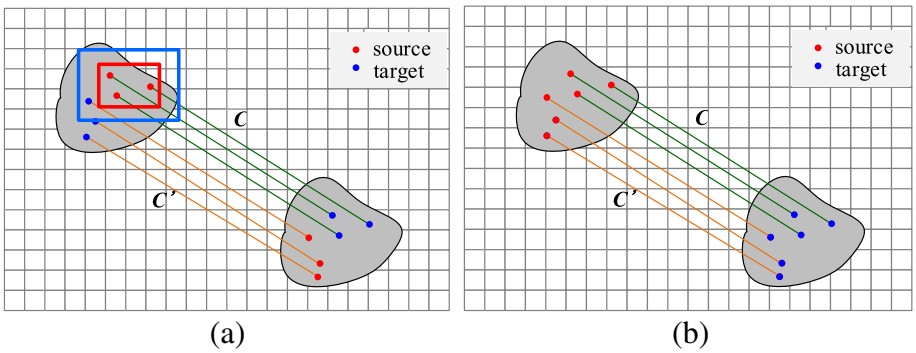

Fig. 4 An example for clustering profile. **a** The matching pairs stay in the opposite subgroup corresponding to group C and group C′ in the same region. **b** Clustering objective

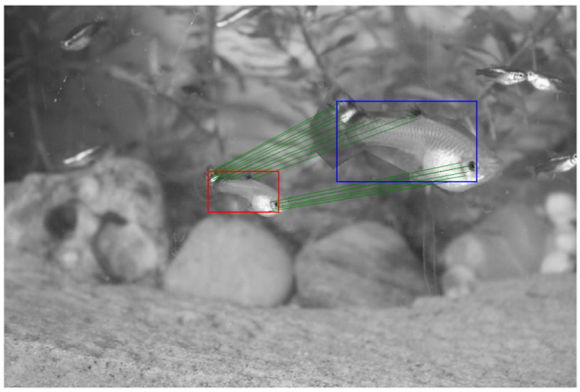

Fig. 5 Clustering and merging results based on our proposed method

$$\frac{D(A,B)}{D(A,C)} \le T_{NNDR}, \tag{2}$$

where D is the Euclidean distance between the descriptors of two keypoints, keypoints A and B are the nearest neighbors, and keypoint A and keypoint C are the second-nearest neighbor. T_{NNDR} is a constant value. If Eq. (2) is satisfied, keypoints A and B are regarded as a matching pair. Generally, keypoint A is the source point and keypoint B is the target point. Our approach uses the Euclidean distance between descriptors to estimate the similarities.

After computing the distances for all keypoints, we can obtain all matching pairs in an image. In order to avoid incorrect matching pairs, if the distance between matched pairs is less than T_{NNDR}, they will be ignored and deleted.

3.2 Clustering and group merging

Our clustering strategy includes clustering and group combining. We improve the clustering method proposed in [16] to perform the coarse clustering process. A clustering yields two match groups: source and destination. They are considered as correspondent regions inside the image and are good cloning. In [16], the clustering strategies only used spatial distance and correspondence angle between matched pairs to perform the clustering. However, when the forgery region is too large, it could result in the matching pairs belonging to the same group that are assigned to the different groups, as shown in Fig. 3. That is, a group may be segmented into many subgroups. In Fig. 3, the red subgroups could not be merged together into a group, and the blue subgroups could not be merged together either.

Hence, in order to solve this problem, we improve the clustering strategies proposed by [16] to achieve the

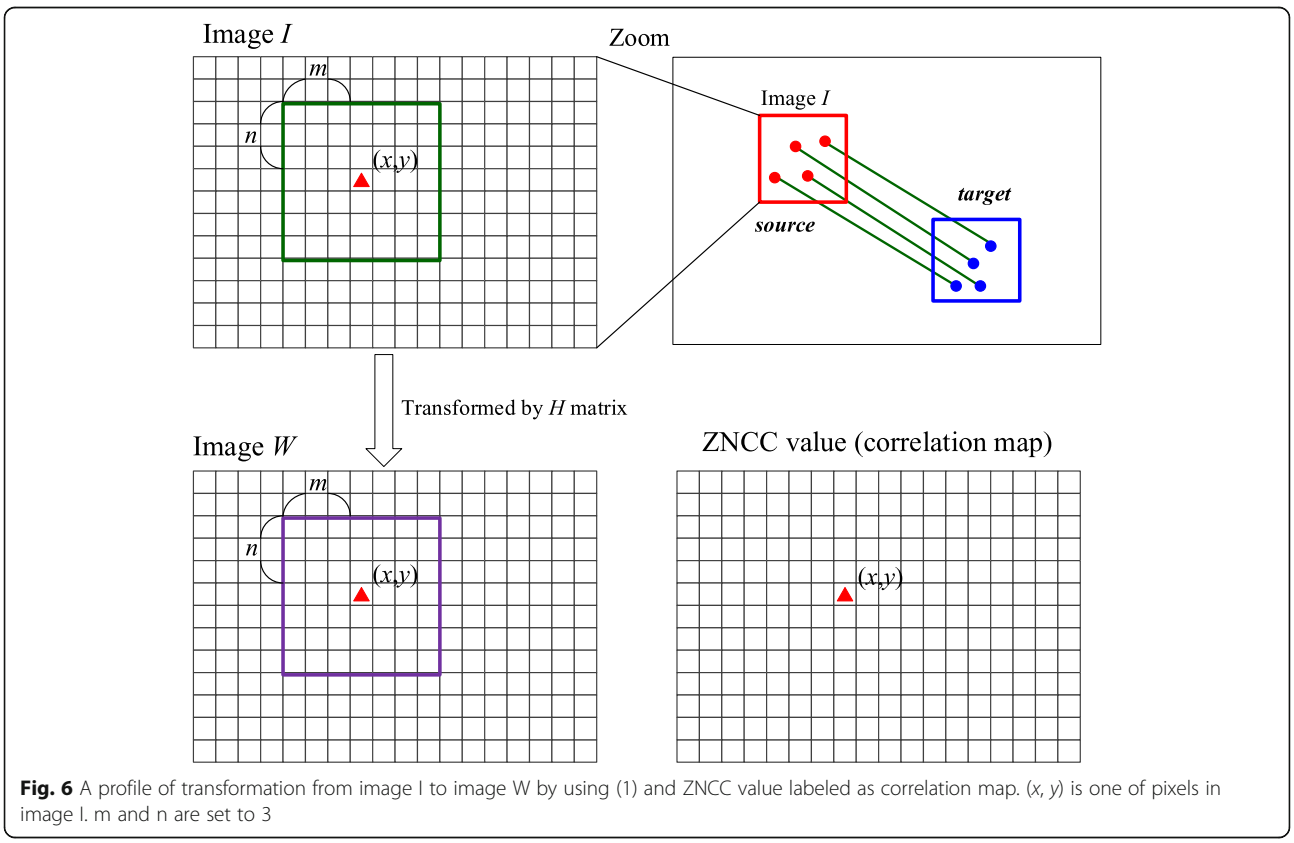

Fig. 6 A profile of transformation from image I to image W by using (1) and ZNCC value labeled as correlation map. (*x*, *y*) is one of pixels in image I. m and n are set to 3

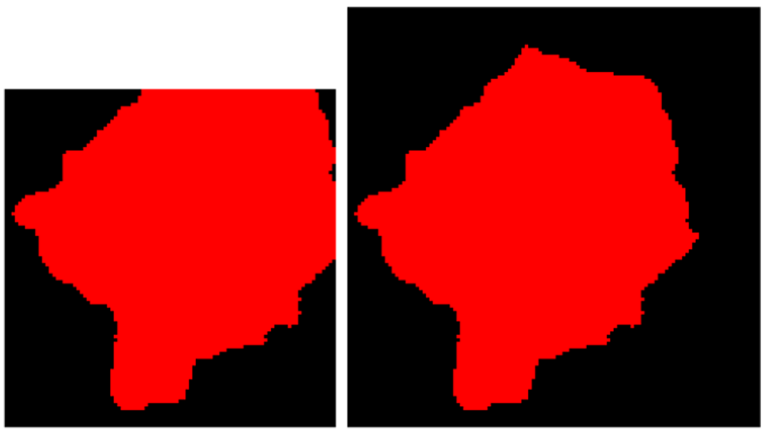

Fig. 7 An example of the binary correlation map and extension. The red color indicates "true" in the correlation map. **a** The red region is the largest region after performing the connected-component operation. This region touched the top border and right border. **b** The result after extending the direction of the top and right borders

coarse clustering. The modified clustering schemes are described by the following. Given any two matching pairs belonging to corresponding subgroups (source and target subgroups), they are considered as correspondent regions in an image and are tampering candidates.

- Spatial adjacency: consider that we have a match pair between keypoints A and B belonging to group G. Keypoint A might belong to the G_{source} subgroup, and keypoint B might belong to the G_{target} subgroup, or vice versa. For a subgroup to admit a paired keypoint as a new member, the spatial distance between the keypoint and its nearest keypoint in such a subgroup needs to be smaller than a predefined threshold, T_c. Moreover, it is necessary to analyze both matched keypoints,

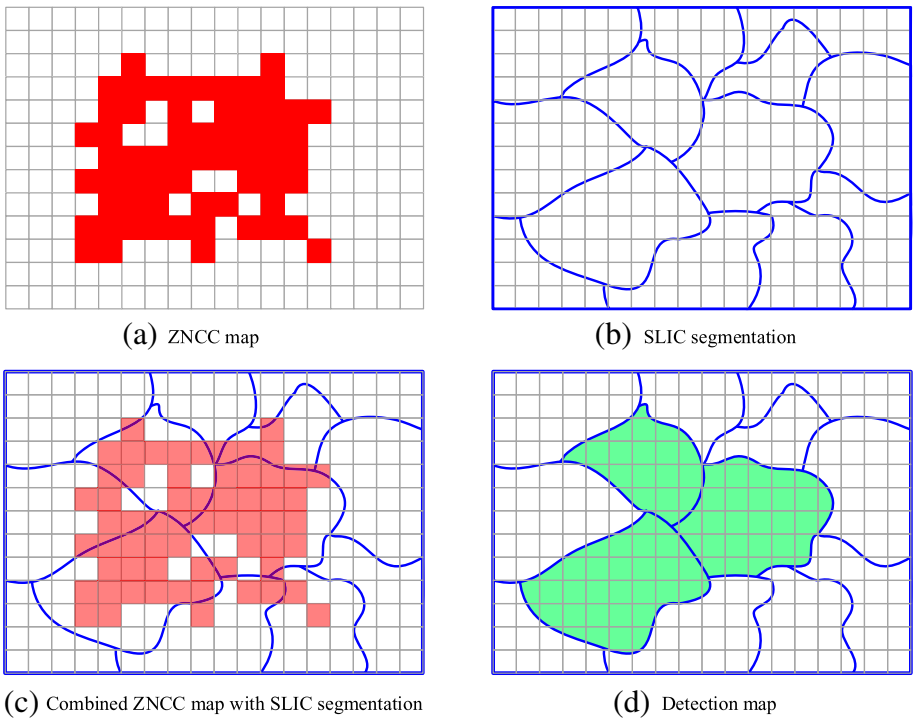

Fig. 8 The profile of the detection map. **a** ZNCC map. **b** SLIC segmentation. **c** Combined ZNCC map with SLIC segmentation. **d** Detection map

Table 1 Setup for the parameters

Parameter	Value
T_{NNDR} [31]: a threshold for nearest neighbor distance	0.6
T_c [16]: a threshold for making a subgroup	50
T_h: a threshold for making a group	$10n$, n: number of keypoints in group.
N_e: the number of the extension	5
R_e: the range of each extension	0.25
S_r: the rectangle search region	0.125
T_k: a threshold for the number of keypoints in a group	5
m, n: the size of computing ZNCC value	7
T_b: a threshold	0.55
E_r: the expanded range	0.25
S: a superpixel size	300
N_d: the number of pixels with true in the ZNCC binary map	0.5

since they have to be in the same group, but in different subgroups.

- The angle consistency: the angle in the range of [0 , 360] with a 15 step is used to determine the angle consistence. It can obtain 24 range partitions. As described above, a new keypoint A candidate to be included into G_{source} will be included in G_{source}, only if the angle of the line that connects the candidate point A and its matching point B stays in the same range of the other points in G_{source}.

After performing coarse clustering, we will further merge these clusters based on the Helmert transformation and spatial adjacent relationship between clusters. Therefore, the transformation can efficiently merge some clusters with a high correlation into a compact cluster. A Helmert transformation is used to describe the relationships between two different coordinate systems without distortion. In 2D space, the Helmert transformation is defined as Eq. (1). We use the Helmet transformation to analyze the geometric relationships between matching pairs. Assuming that the number of keypoints in a cluster is greater than one, we will compute the Helmert parameters of the cluster (source and

Table 2 Detected results for CMH1 dataset under simple copies

Method	Recall (%)	Precision (%)	FPR (%)	F_1 (%)
Proposed	96.50	97.66	2.31	97.08
Amerini et al. [13]	93.57	94.52	5.41	94.04
Silva et al. [16]	92.34	97.88	2.0	95.03
Pun et al. [18]	92.00	92.93	6.99	92.46
Li et al. [19]	98.17	57.64	72.14	72.63

Table 3 Detected results for D0 dataset under simple copies

Method	Recall (%)	Precision (%)	FPR (%)	F_1 (%)
Proposed	84.88	92.81	3.39	88.67
Amerini et al. [13]	73.41	89.38	2.42	80.61
Silva et al. [16]	64.14	82.02	1.89	71.99
Pun et al. [18]	62.08	82.32	1.72	70.78
Li et al. [19]	77.09	64.19	49.42	70.05

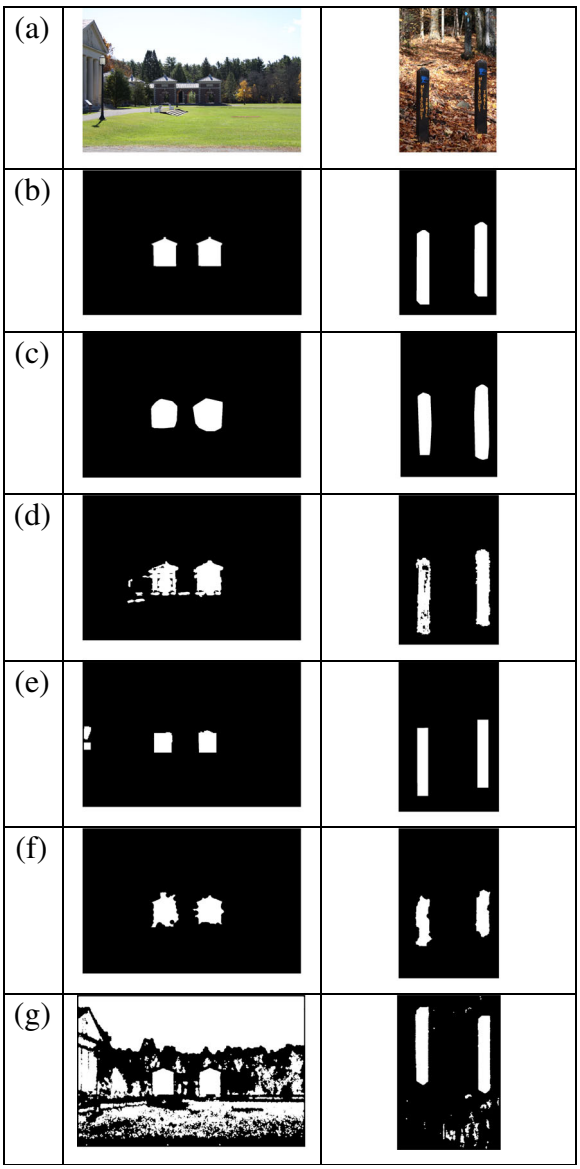

Fig. 9 Detection results under simply copy-move. **a** Forgery images. **b** Ground truth. **c** Our proposed. **d** Amerini et al. [13]. **e** Silva et al. [16]. **f** Pun et al. [18]. **g** Li et al. [19]

Table 4 Detected results for CMHD2 dataset under rotation

Method	Recall (%)	Precision (%)	FPR (%)	F_1 (%)
Proposed	97.06	98.81	1.17	97.93
Amerini et al. [13]	79.89	96.58	2.83	87.45
Silva et al. [16]	66.59	98.89	0.75	79.59
Pun et al. [18]	86.04	97.84	1.90	91.56
Li et al. [19]	63.89	57.58	47.06	60.57

Table 5 Detected results for D1 dataset under rotation

Method	Recall (%)	Precision (%)	FPR (%)	F_1 (%)
Proposed	92.19	97.36	1.12	94.70
Amerini et al. [13]	79.20	91.11	0.47	84.74
Silva et al. [16]	55.47	79.34	0.71	65.29
Pun et al. [18]	53.84	86.17	0.57	66.27
Li et al. [19]	24.08	25.08	52.23	24.57

target subgroup); otherwise, this cluster will be discarded. For instance, given any two matching pairs, by assuming that (X_p, Y_p) are target coordinates and (x_p, y_p) are source coordinates, the transformation can easily compute and obtain four Helmert parameters by Eq. (1).

Assuming that there is a keypoint from another group, C', within the search range we specified, this keypoint will be checked whether it belongs to source or target subgroup. It is because we do not constrain which keypoint stays in source or target subgroup for a matching pair in the previous matching process. During the matching process, the same region may be clustered into different groups, and the matching pairs may stay in the subgroup opposite to the other, as shown in Fig. 4a. Assuming that this keypoint belongs to the target subgroup in group C', we will exchange all members in the target subgroup with those of the source subgroup in group C', as shown in Fig. 4b. Afterwards, we transform all members in the source subgroup for group C' to new members in target subgroup by means of the Helmert parameters derived from group C. Then, we will compute the difference in the spatial coordinates between target keypoints in group C and new target keypoints in group C'. When this difference is smaller than a threshold, T_h, two groups are merged and then Helmert parameters derived from group C are updated. Based on our experimental test, we assigned the threshold value, $T_h = 10n$, where n denotes the number of keypoints in the group.

Next, we use a rectangular search range, which is defined as $(x_{\max}, y_{\max}, x_{\min}, y_{\min})$ belonging to the lower right and upper left coordinates of keypoints in source subgroup, to perform group merging. The target subgroup also creates a rectangular search range**.** If there is no keypoint presented in the rectangular search range, this rectangular range will expand the search range to find other clusters until one of the terminal conditions is satisfied. The terminal conditions are defined as follows.

1. The number of the extension (N_e) has reached a value of five, and there is no cluster that can be combined. Here, the range of each extension (R_e) is multiplied the rectangle searching region by a factor of 1.25.
2. The rectangle search region (S_r) is greater than 0.125 times of size of a host image.

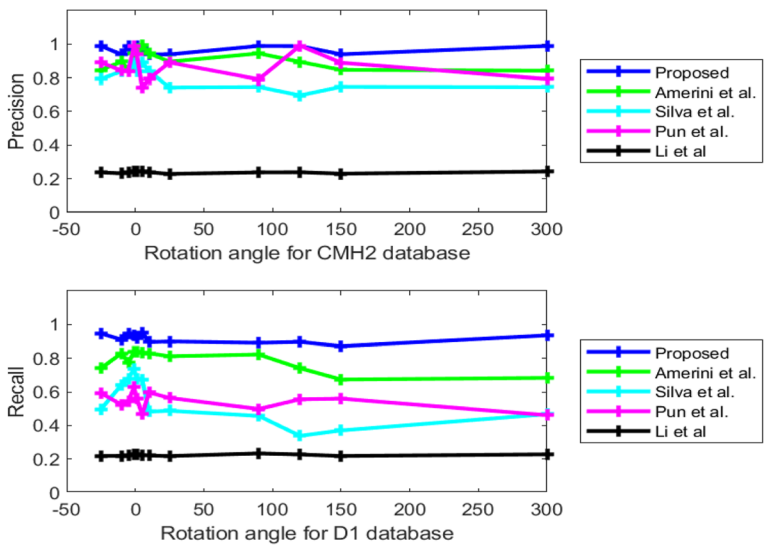

Fig. 10 The precision and recall values under the different rotation angles compared to our proposed method, Amerini et al.'s method, Silva et al.'s method, Pun et al.'s method, and Li et al. method

Repeat the above steps until no clusters can be combined. Finally, we remove the invalid clusters that involve less than five keypoints.

After performing the merging process, some clusters can be integrated into a larger cluster, as shown in Fig. 5.

3.3 Forgery regions localization and refining

We use zero mean normalized cross-correlation (ZNCC) [32] to measure the similarity between source regions and target regions. Assuming that a Helmert transform matrix, H, exists, the relationships between source group and target group are expressed as [13]. Let $a = [x_1 \ y_1]^T$ be a point in the source cluster and $b = [x_2 \quad y_2]^T$ be a point in the target cluster, then $a = H \times b$, and since H is invertible, $b = H^{-1} \times a$. Combining these relationships and the ZNCC measurement, the forgery region can be further localized.

First, if the number of keypoints in a group is less than a threshold (T_k), we will regard this group as unimportant and it will be discarded. The source subgroup with the matching points for each cluster is labeled as image I, and all the pixel points (x, y) in image I are converted to the new locations (x', y') in image W by using the Helmert parameters defined as Eq. (1). Therefore, a new image W of the same size is produced. Then, we create a ZNCC binary map using Eq. (3).

In order to obtain the similarity and distance between image I and image W, we compute the ZNCC. In addition, we also define a correlation map to record the similarity between image I and image W. The ZNCC is defined as

$$ZNCC(x,y) = \frac{\sum_{j=-n}^{n}\sum_{i=-m}^{m}\left(I(x+i,y+j)-\overline{I}\right)\left(W(x+i,y+j)-\overline{W}\right)}{\sqrt{\sum_{j=-n}^{n}\sum_{i=-m}^{m}\left(I(x+i,y+j)-\overline{I}\right)^2\sum_{j=0}^{n-1}\sum_{i=0}^{m-1}\left(W(x+i,y+j)-\overline{W}\right)^2}}, \tag{3}$$

where $I(x, y)$ and $W(x, y)$ are the gray-level values at pixel (x, y) in I and W respectively, $\overline{I}$ and $\overline{W}$ are the mean gray-level values around pixel (x, y) in the I and W, respectively, and m and n are the size of neighboring area centered at pixel (x, y). This distance range is the interval [− 1, 1] (1 indicates a perfect match, and 0 for "no correlation"). Figure 6 shows a profile for the transformation from image I to image W by using (1) and the ZNCC value (correlation map) by using (3). (x,y) is one of pixels in Image I, and m and n are set to 3.

Then, we apply a Gaussian filter to the correlation map in order to reduce the noisy pixels, and a binary correlation map is given by means of a threshold (T_b). If the ZNCC value for point (x, y) is greater than a threshold, this point (x, y) is assigned as true; otherwise, this point is assigned as false. Next, we will perform connected-component labeling on this binary map. This threshold, T_b, is set to 0.55, which is a value obtained through experimentation.

If the largest region involved in connected-component labeling touches the border of the binary map, it means that the range of this region is bigger than the range of the binary map, as shown in Fig. 7a. The top and right sections of this region touch the borders. Therefore, this region will be expanded in a rectangular interval along the touched border. The steps described above are repeated until the largest region does not touch the border, as shown in Fig. 7b. Based on an empirical value obtained in

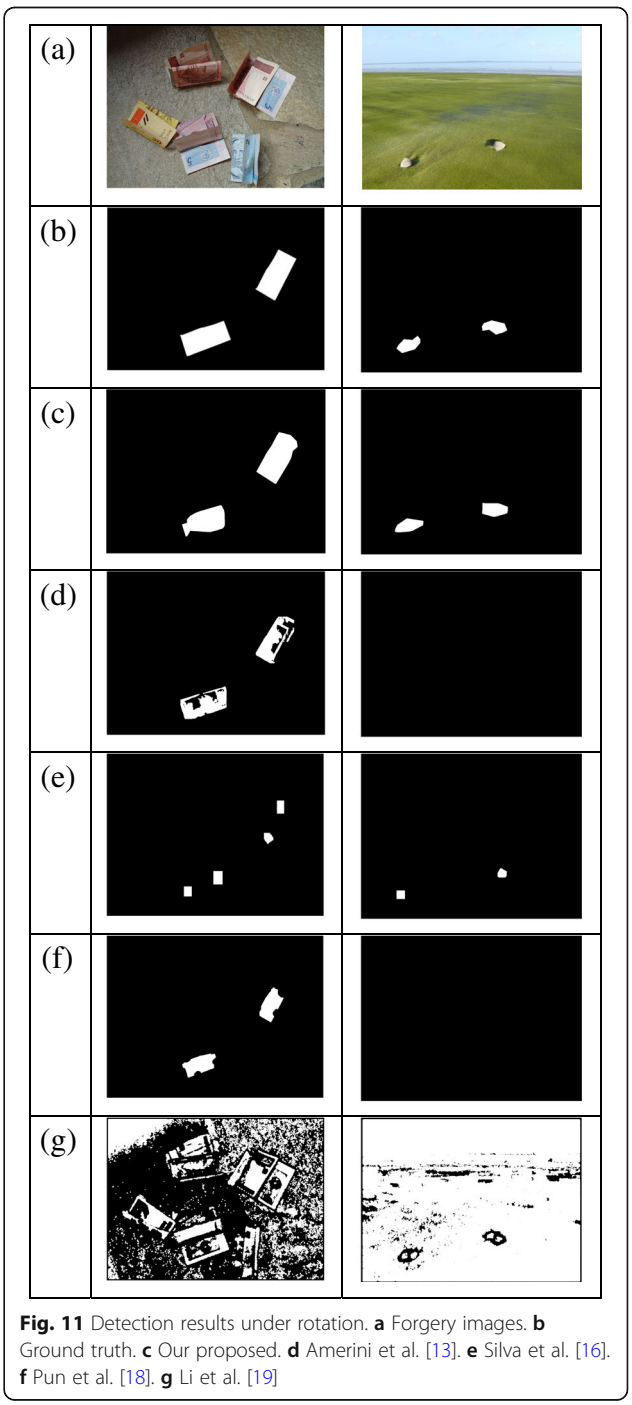

Fig. 11 Detection results under rotation. **a** Forgery images. **b** Ground truth. **c** Our proposed. **d** Amerini et al. [13]. **e** Silva et al. [16]. **f** Pun et al. [18]. **g** Li et al. [19]

Table 6 Detected results for CMH3 dataset under resizing

Method	Recall (%)	Precision (%)	FPR (%)	F_1 (%)
Proposed	86.21	97.53	2.18	91.52
Amerini et al. [13]	77.61	96.89	2.49	86.18
Silva et al. [16]	62.30	97.04	1.90	75.88
Pun et al. [18]	75.14	96.70	2.56	84.57
Li et al. [19]	67.07	70.71	27.78	68.84

our experiments, the expanded range (E_r) is multiplied the width or height of this sub-image by a factor of 1.25 depending on the direction of touching border. All points in image I are finished, the content of the binary correlation map is filled to the ZNCC binary map corresponding to the location. For instance, Fig. 8a shows the ZNCC binary map. Next, we combine the SLIC superpixel segmentation described in Section 2.1 to achieve the forgery region localization.

The host image is segmented into many sub-regions by the SLIC algorithm. In the SLIC algorithm, the smaller the size of a superpixel (S), the greater the number of superpixels present. Moreover, very few true edges are missed. In contrast to increasing size, the number of superpixels is reduced, and many true edges will be missed. Therefore, in our approach, the size of a superpixel (S) is assigned to 300 pixels by experiments. For each sub-region, we will count the number of pixels that are considered true in the ZNCC binary map. If this number (N_d) is greater than a threshold in the relative sub-region, all the pixels in this sub-region are labeled as a detection map that serves as a part of forgery regions, as shown in green color areas of Fig. 8d. Afterwards, we label the connected components as the detection map, and delete the regions that have an area less than 0.1%. Finally, each of the remaining regions will use the convex-hull morphologic method to connect together in the binary detection map. Figure 8 illustrates the profile of the detection map. After performing our proposed method, we can efficiently detect and localize the forgery regions more precisely.

4 Experimental results and discussion

To verify the performance of the proposed image forensics, the experimental results are compared to Amerini et al. [13], Silva et al. [16], Pun et al. [18], and Li et al. [19] to

Table 7 Detected results for D2 dataset under resizing

Method	Recall (%)	Precision (%)	FPR (%)	F_1 (%)
Proposed	79.09	86.76	0.99	82.75
Amerini et al. [13]	61.58	72.27	0.47	66.50
Silva et al. [16]	37.31	60.89	0.52	46.27
Pun et al. [18]	38.93	72.10	0.40	50.56
Li et al. [19]	26.82	28.17	52.25	27.48

Table 8 Detected results for CMH4 dataset under resizing and rotation

Method	Recall (%)	Precision (%)	FPR (%)	F_1 (%)
Proposed	87.08	97.10	2.26	91.82
Amerini et al. [13]	76.80	93.55	5.30	84.35
Silva et al. [16]	63.72	97.64	1.54	77.11
Pun et al. [18]	71.56	97.96	1.49	82.7
Li et al. [19]	63.33	56.89	47.98	59.94

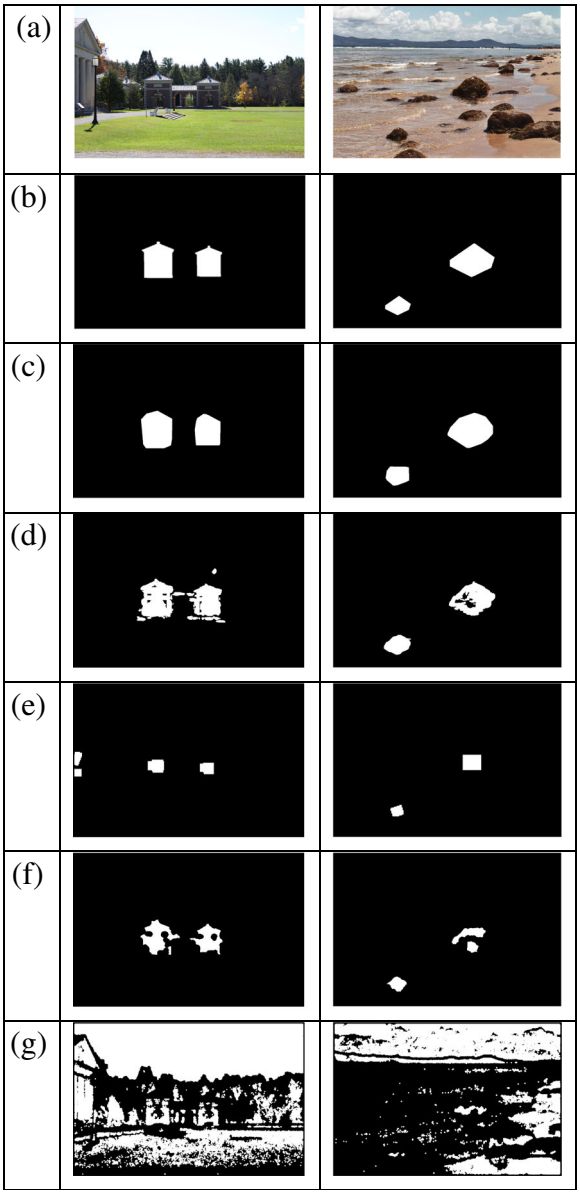

Fig. 12 Detection results under resizing. **a** Forgery images. **b** Ground truth. **c** Our proposed. **d** Amerini et al. [13]. **e** Silva et al. [16]. **f** Pun et al. [18]. **g** Li et al. [19]

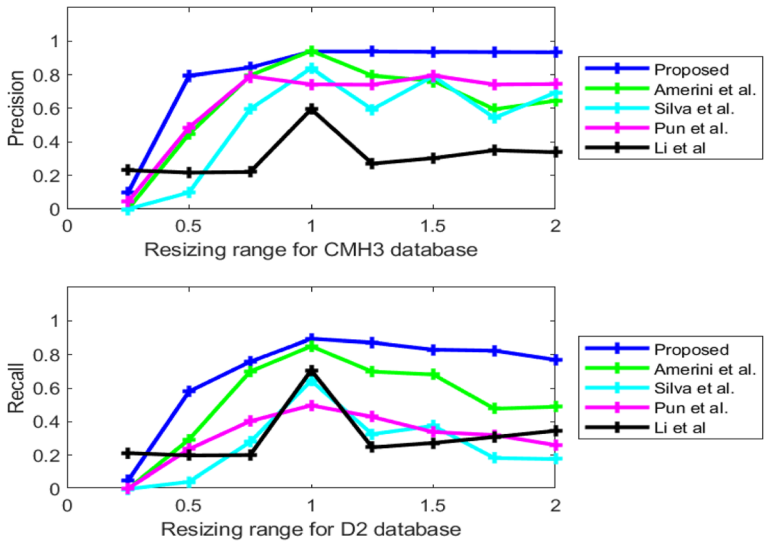

Fig. 13 The precision and recall values under the different resizing ranges compared to our proposed method, and the methods of Amerini et al., Silva et al., Pun et al., and Li et al.

perform the forgeries, including copying and translations, scaling, rotation, and compression.

4.1 Experimental setup and datasets

Table 1 illustrates the parameters presented in the experiments. According to our experiments, we systematically vary the related thresholds within 50% to 200% and observe performance changes; afterwards, they are given, and some thresholds are derived from the literature [16, 31]. However, the assignment of these parameter values can be modified by the user based on the data. The experiments were implemented in Microsoft Visual Studio C#, on an Intel® core i5–4570@ 3.2 GHz computer with 4 GB of RAM running a Windows 7 64 bits platform.

We used nine public datasets for our demonstration. CMH1–4 datasets and a compressed dataset (CMH5) constructed by [16] contain sizes varying from 845×634 pixels (the smallest) to 1296×975 pixels (the biggest) and are stored in the PNG format. The D0–3 datasets constructed by [17] contain sizes of 1000×700 pixels or 700×1000 pixels and stored in the BMP format. The CMH series datasets depict as follows.

- CMH1: 23 images that were only copied and then translated.
- CMH2: 25 images with a rotation of the duplicated region, the orientations are in the range of [−90 , 180].
- CHM3: 26 images with resizing of the duplicated region; the scaling range is between 80 and 154%.
- CHM4: 34 images with rotation and resizing entirely.
- CMH5: 108 images that are derived from 36 randomly selected images from the CMH1–4 datasets and compressed with a quality factor of 70%, 80%, 90%.

The D0–3 datasets depict as follows.

- D0 dataset: 50 images that are copied and translated.
- D1 dataset: 600 images with a rotation of the duplicated region. This dataset is further subdivided into subsets. The first subset, D1.1, is created by rotating the copies with 11 different types of rotation around the angle zero in the range of [−25 , 25] with a step of 5 . The second subset, D1.2, is created by rotating the copies with 12 different types in the range of [0 , 360] with a step of 30 . The third subset, D1.3, is created by rotating the copies with 11 different types in the range of [−5 , 5] with a step of 1
- D2 dataset: 320 images with resizing of the copied region. This dataset is subdivided into two subsets. The D2.1 subset is obtained by scaling the copies with 8 different scaling factors in the range of [0.25, 2] with a step of 0.25. The D2.2 subset is scaled by 11 scaling factors in the range of [0.75, 1.25] with a step of 0.05.
- D3 dataset: 50 original images without tampering to verify the forensic ability between tampered and untampered images.

Every image in every dataset has its own binary ground truth displaying the original and duplicated regions in white

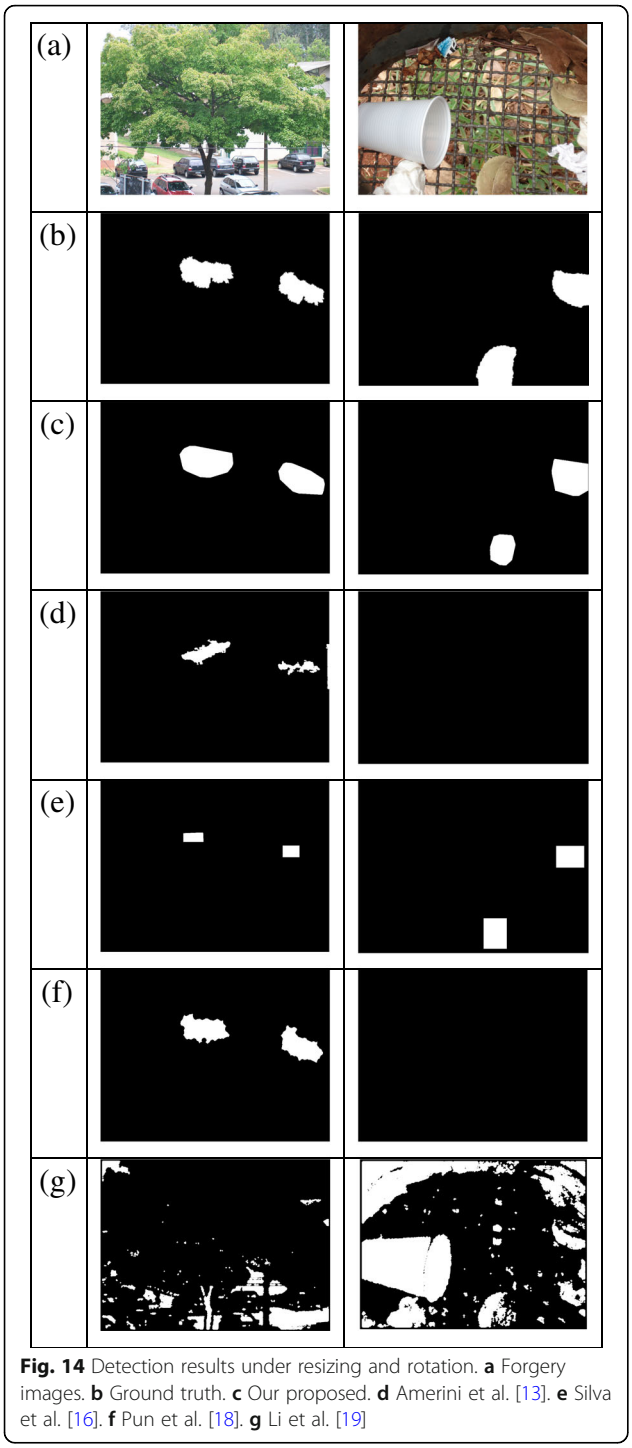

Fig. 14 Detection results under resizing and rotation. **a** Forgery images. **b** Ground truth. **c** Our proposed. **d** Amerini et al. [13]. **e** Silva et al. [16]. **f** Pun et al. [18]. **g** Li et al. [19]

color. And the tampered region within the datasets is of a single region copied one time and stayed in the same image.

4.2 Performance evaluation

For performance evaluation, we used the precision, recall, F_1 [8, 18], and the false positive rate (FPR) [16] to demonstrate our proposed method. These evaluation criteria are expressed as:

Table 9 Detected results for JPEG compression

Method	Recall (%)	Precision (%)	FPR (%)	F_1 (%)
Proposed	68.51	78.45	1.22	73.14
Amerini et al. [13]	45.59	61.70	3.65	52.44
Silva et al. [16]	31.19	64.54	0.65	42.06
Pun et al. [18]	45.76	69.69	2.00	55.24
Li et al. [19]	31.35	31.16	48.80	31.25

- *Precision*: represents the probability that the detected regions are truly the forgery regions, as expressed in (4).

$$\text{precision} = \frac{|\text{TP}|}{|\Omega_{\text{retrieved}}|} \tag{4}$$

where $|\Omega_{\text{retrieved}}|$ denotes the number of the detected forgery pixels by our proposed method from the datasets, $|\text{TP}|$ (true positive) represents the number of correctly detected forged pixels labeled as forged regions in the ground truth.

- *Recall*: represents the probability that the forgery regions are detected, as expressed in (5).

$$\text{recall} = \frac{|\text{TP}|}{|\Omega_{\text{relevant}}|}. \tag{5}$$

where $|\Omega_{\text{relevant}}|$ represents the ground truth forgery regions of the datasets.

- F_1: this score combines both the precision and recall into a signal value. It is calculated by (6).

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{6}$$

- FPR: indicates the percentage of incorrectly located tampering regions. It is defined as

$$\text{FPR} = \frac{|\text{FP}|}{|\Omega_{\text{normal}}|}, \tag{7}$$

where $|\Omega_{\text{normal}}|$ represents the number of pixels that do

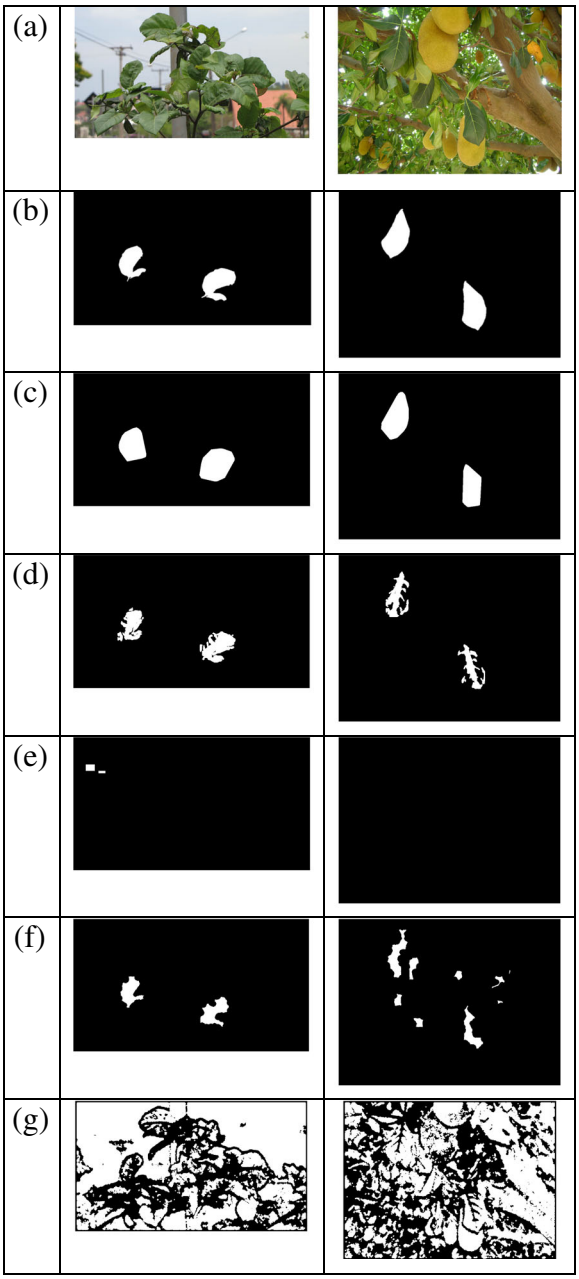

Fig. 15 Detection results under compression (CMH5 dataset). **a** Forgery images. **b** Ground truth. **c** Our proposed. **d** Amerini et al. [13]. **e** Silva et al. [16]. **f** Pun et al. [18]. **g** Li et al. [19]

Table 10 Detected results for D3 dataset

Method	Proposed	Amerini et al. [13]	Silva et al. [16]	Pun et al. [18]	Li et al. [19]
FPR (%)	0.05	0.12	0.39	2.34	48.97

not belong to the tampering regions in the ground truth, |FP| (false positive) denotes the number of wrongly detected as tampering pixels by our proposed method.

Because the datasets have been tampered with in different ways, they are not consistent in our experiments, and therefore we compute the average values for these evaluation criteria in the dataset to verify the performance. As indicated above, the *precision* is the probability that a detected forgery is truly a forgery, and the *recall* is the probability that a forgery image is detected. Generally, a higher *precision* and a higher *recall* represent better performance.

5 Results

Regarding the different forgery images created by copying and translation, scaling, rotation, and compression, the experimental results are presented and discussed in the following section.

5.1 Detection results for copying and translation

The forgery images are simply copied and moved operations, such as the CMH1 and D0 datasets. Tables 2 and 3 illustrate the detected results compared to our proposed method and the methods of Amerini et al., Silva et al., Pun et al., and Li et al.. Figure 9 presents several detection results for simple copying.

5.2 Detection results for rotation

Tables 4 and 5 illustrate the detection results under rotation forgery for the CMH2 and D1 datasets. Figure 10 presents the precision/recall results under rotation with the different angles.

As shown in Table 4, it is evident that our proposed method can achieve a higher precision (98.81%), and at the same time, can gain a much better recall (97.06%) and F_1 (97.73%). From Table 5, it is obvious that our approach can achieve precision (97.36%), recall (92.18%), and F_1 (94.70%). The results obtained by our proposed method are much better comparative methods. Figure 11 shows some detection results for rotation tampering.

As presented above, our proposed method can achieve much better detection results under rotation forgery.

5.3 Detection results for resizing/scaling and rotation

Tables 6 and 7 present a comparison of results under resizing forgery for the CMH3 and D2 datasets. Table 8 illustrates the detection results under resizing and rotation operations using the CMH4 dataset. Figure 12 presents several detection results for resizing forgeries.

Regarding the detection results under resizing forgery, as shown in Tables 6 and 7, it is obvious that the precision and recall of our proposed method are superior to comparable methods. The precision value can maintain 86.76% and the recall value can maintain 79.09%. Figure 13 illustrates the precision/recall results under the

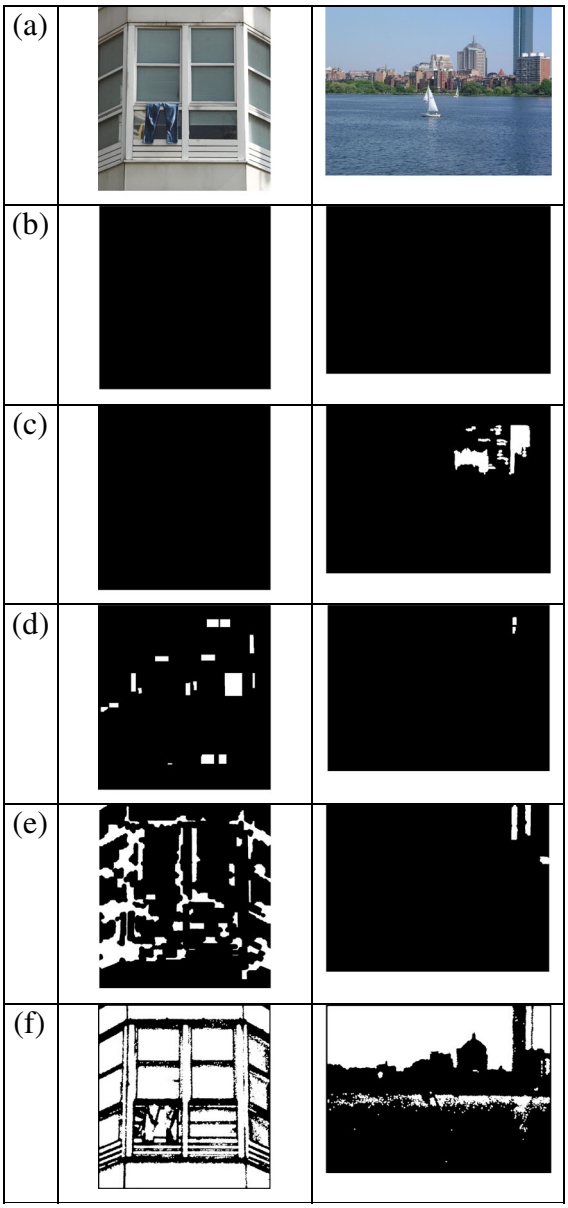

Fig. 16 Detection results under untampered. **a** Forgery images. **b** Our proposed. **c** Amerini et al. [13]. **d** Silva et al. [16]. **e** Pun et al. [18]. **f** Li et al. [19]

Table 11 Average execution time per image in second

	CMH series	D series
Our proposed	261.29	263.75
Amerini et al. [13]	257.95	277.72
Silva et al. [16]	63.53	58.43
Pun et al. [18]	204.84	199.18
Li et al. [19]	428.99	381.77

different scaling ranges. In addition, for resizing and rotation forgeries, we can still keep a higher precision and the best recall, compared to the methods used by Amerini et al., Silva et al., Pun et al., and Li et al., as shown in Table 8. Figure 14 shows the detection results under resizing and rotation forgeries.

5.4 Detection results for compression

We used a CMH5 [16] to perform the forgery detection. A comparison of the results is shown in Table 9 and Fig. 15.

From Table 9, it is obvious that our proposed results can achieve the best precision and recall results, and are superior than that of the methods of Amerini et al., Silva et al., Pun et al., and Li et al.

In order to estimate the robustness of our approach against false positive detection, we used an untampered dataset (D3 dataset) to verify our approach. Table 10 illustrates the FPR result. From Table 10, the FPR value of our proposed method is the lowest compared to other methods. Figure 16 shows some detection results under un-tampering images. From these results, it is obvious that our proposed method can efficiently decide whether the host image has been tampered with or not.

5.5 Comparison of execution time

In this section, execution times are illustrated for both datasets. The average execution times are illustrated in Table 11. Generally, all methods need more time on both datasets due to the high resolution of images, except Silva et al. [16]. This is because the method used by Silva et al. utilized the V channel of the HSV color space to extract a low number of SURF keypoints. Owing to the ZNCC used in our proposed method, the execution time of our proposed method is slightly slower than that of the method used by Pun et al.

In summary, the proposed method can achieve impressive accuracy in a reasonable time.

6 Discussions

The experimental results have been presented above. In summary, for CMH1–4 datasets including copying and translations, rotation, resizing/scaling, resizing, and rotation, the precision values are greater than 97.1%, and the recall values achieved at least 86.21%. The global performance of the precision/recall values is superior to the methods used by Amerini et al., Silva et al., Pun et al., and Li et al., as shown in Tables 2, 4, 6, and 8. For the D0–2 datasets, the precision values can achieve 97.36% (the highest) and 86.76% (the lowest) and the recall values can achieve 92.18% (the highest) and 79.09% (the lowest), as shown in Tables 3, 5, and 7. It is clear that the detection results for the D0–2 datasets obtained by our approach

are superior to that obtained by the methods of Amerini et al., Silva et al., Pun et al., and Li et al. At the same time, for compression forgery, Fig. 15 and Table 9 show the results compared to the other methods. For robustness of false positive detection, our approach has sufficient robustness to resist the false detection, and Fig. 16 and Table 10 present the results. For execution time, although our approach is not the fastest, it is acceptable. For rotation tampering, our proposed method can detect copy-move forgery regions with a large rotation. Evidently, our proposed system works well for image forensics under rotation, scaling, and compression forgery attacks, compared to other methods.

7 Conclusions

In this study, the major strategy of our proposed algorithm focuses on a single tampered region detection. And we have proposed keypoint-based image forensics for copy-move forgery images based on a Helmert transformation and SLIC superpixel segmentation. Compared to the sliding window approach, the keypoint-based technique can be applied at a lower computational cost because of the significantly reduced number of points required. In addition, we use the Helmert transformation to estimate the geometric relationships between matching pairs and to work the merging clusters. On the other hand, we use an SLIC algorithm to localize the tampering regions more precisely. Based on these strategies, we can keep much more important information to conduct image forensics.

As previously presented in the experiments, it is clear that the proposed method is highly robust against many kinds of forged images, such as geometric transformations (scaling, rotation) and JPEG compression. However, the current method is not robust against symmetric, recurring, and smooth patterns for tampering region. Progress in detecting symmetric, recurring, smooth forgery images, and tampering region copied multiple times will be a major focus in the future.

Abbreviations

CIELAB: International Commission on Illumination, *L* for the lightness and *A* and *B* for the green–red and blue–yellow color components; DCT: Discrete cosine transform; DWT: Discrete wavelet transform; FPR: False positive rate; HAC: Hierarchical agglomerative clustering; HOG: Histogram of oriented gradient; LPM: log-polar maps; NNDR: Nearest neighbor distance ratio; SIFT: Scale-invariant feature transform; SLIC: Simple linear iterative clustering; SURF: Speedup robust features; TPR: True positive rate; ZNCC: Zero mean normalized cross-correlation

Acknowledgements

Not applicable.

Authors' contributions

For corresponding author: Dr. Huang is major to design the framework of the system and theorem base and writing.
Author 2: Miss Ciou is to code and implement the experimental results.

Funding

There are not any funding for my affiliation, but I will pay an article publication fee by myself if my manuscript is accepted.

Availability of data and materials

The test sets can take from [16, 17].

Competing interests

The authors declare that they have no competing interests in this work.

Received: 11 October 2018 Accepted: 28 May 2019
Published online: 13 June 2019

References

1. M. Al-Qershi Osamah, B.E. Khoo, Passive detection of copy-move forgery in digital images: state-of-the-art. Forensic Sci. Int. **23**(1), 284–295 (2013)
2. J.W.G. Wang, J. Liu, A. Zhang, Y.W. Dai, Z.Q. Wang, Fast and robust forensics or image region-duplication forgery. Acta Automat., Sinica **35**(12), 1488–1495 (2009)
3. S.J. Ryu, M.J. Lee, H.K. Lee, in *Information Hiding*. Detection of copy-rotate-move forgery using Zernike moments (Springer-Verlag, Berlin, 2010), pp. 51–65
4. S.J. Ryu, M. Kirchner, M.J. Lee, H.K. Lee, Rotation invariant localization of duplicated image regions based on Zernike moments. IEEE Trans. Inf. Forensics Secur. **8**(8), 1355–1370 (2013)
5. Y. Huang, W. Lu, W. Sun, D. Long, Improved DCT-based detection of copy-move forgery in images. Forensic Sci. Int. **206**(3), 178–184 (2011)
6. X. Wang, X. Zhang, Z. Li, S. Wang, in *3rd Int. Conf. on Multimedia Information Networking and Security*. A DWT-DCT based passive forensics method for copy-move attacks (2011), pp. 304–308
7. S. Bravo-Solorio, A.K. Nandi, Automated detection and localisation of duplicated regions affected by reflection, rotation and scaling in image forensics. Signal Process. **91**(8), 1759–1770 (2011)
8. R. Davarazni, K. Yaghmaie, S. Mozaffari, Copy-move forgery detection using multiresolution location binary patterns. Forensic Sci. Int. **231**(1), 61–72 (2013)
9. J.C. Lee, C.P. Chang, W.K. Chen, Detection of copy-move image forgery using histogram of orientated gradients. Inf. Sci. **321**, 250–262 (2015)
10. L. Li, S. Li, H. Zhu, X. Wu, Detecting copy-move forgery under affine transforms for image forensics. Comput. Electr. Eng. **40**(6), 1951–1962 (2014)
11. X. Pan, S. Lyu, Region duplication detection using image feature matching. IEEE Trans. Inf. Forensics Secur. **5**(4), 875–867 (2010)
12. I. Amerini, L. Ballan, R. Caldelli, A.D. Bimbo, G. Serra, A sift-based forensic method for copy-move attack detection and transformation recovery. IEEE Trans. Inf. Forensics Secur. **6**(3), 1099–1110 (2011)
13. I. Amerini, L. Ballan, R. Caldelli, A.D. Bimbo, D. Tongo, G. Serra, Copy-move forgery detection and localization by means of robust clustering with J-linkage. Signal Process. Image Commun. **28**(6), 659–669 (2013)
14. B.L. Shivakumar, S.S. Baboo, Detection of region duplication forgery in digital images using SURF. Int. J. Comput. Sci. **8**(1), 199–205 (2011)
15. P. Mishra, N. Mishra, S. Sharma, R. Patel, Region duplication forgery detection technique based on SURF and HAC. Sci. World J. **2013**, 1–8 (2013)
16. 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. J. Vis. Commun. Image Represent. **29**, 16–32 (2015)
17. E. Ardizzone, A. Bruno, G. Mazzola, Copy-move forgery detection by matching triangles of keypoints. IEEE Trans. Inf. Forensics Secur. **10**(10), 2084–2094 (2015)
18. C.M. Pun, X.C. Yuan, X.L. Bi, Image forgery detection using adaptive over-segmentation and feature points matching. IEEE Trans. Inf. Forensics Secur. **10**(8), 1705–1716 (2015)
19. J. Li, X. Li, B. Yang, X. Sun, Segmentation-based image copy-move forgery detection scheme. IEEE Trans. Inf. Forensics Secur. **10**(3), 507–518 (2015)
20. D. Cozzolino, G. Poggi, L. Verdoliva, Efficient dense-field copy-move forgery detection. IEEE Trans. Inf. Forensics Secur. **10**(11), 2284–2297 (2015)
21. D. Chauhan, D. Kasat, S. Jain, V. Thakare, in *Int. Conf. On computational modelling and security (CMS2016)*. Survey of keypoints based copy-move forgery detection methods on image (2016), pp. 206–212
22. D.G. Lowe, Distinctive image features from scale-invariant keypoint. Int. J. Comput. Vis. **60**(2), 91–110 (2004)

23. H. Bay, T. Tuytelaars, L. Van Gool, in *Computer Vision-ECCV2006*. SURF: speeded up robust features (Springer-Verlag, Berlin, 2006), pp. 404–417
24. N. Warif, A. Wahab, M. Idris, R. Ramli, R. Salleh, S. Shamshirband, K. Choo, Copy-move forgery detection: survey, challenges and future directions. J. Netw. Comput. Appl. **75**, 259–278 (2016)
25. M. Zandi, A. Mahmoudi-Aznaveh, A. Talebopur, Iterative copy-move forgery detection based on a new interest point detector. IEEE Trans. Inf. Forensics Secur. **11**(11), 2499–2512 (2016)
26. M. Ikhalyel, M. Hariadi, K.E. Pummama, A study of copy-move forgery detection based on segmentation. IJCSNS Int. J. Comp. Sci. Netw. Secur. **18**(7), 27–32 (2018)
27. J. Zheng, Y. Liu, J. Ren, T. Zhu, Y. Yan, H. Yabf, Fusion of block and keypoints based approaches for effective copy-move image forgery detection. Multidim. Syst. Sign. Process. **24**(4), 989–1005 (2016)
28. V. Christlein, C. Riess, J. Jordan, C. Riess, E. Angelopoulou, An evaluation of popular copy-move forgery detection approaches. IEEE Trans. Inf. Forensics Secur. **7**(6), 1841–1854 (2012)
29. R. Achanta, A. Shaji, K. Simth, A. Lucchi, P. Fua, S. Süsstrunk, SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2274–2282 (2012)
30. R.E. Deakin, Coordinate transformations in surveying and mapping. Geospat. Sci. (2004) *[Online]. available* http://www.mygeodesy.id.au/documents/COTRAN_1.pdf
31. K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. **27**(10), 1615–1630 (2005)
32. L.D. Stefano, S. Mattoccia, F. Tombari, ZNCC-based template matching using bounded partial correlation. Pattern Recogn. Lett. **26**(14), 2129–2134 (2005)

8 Publisher's Note