J. Vis. Commun. Image R. 29 (2015) 16–32

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Going deeper into copy-move forgery detection: Exploring image telltales via multi-scale analysis and voting processes ☆

Ewerton Silva, Tiago Carvalho *, Anselmo Ferreira, Anderson Rocha

RECOD Lab., Institute of Computing, University of Campinas, Av. Albert Einstein, 1251, Cidade Universitária "Zeferino Vaz", Campinas, SP 13083-852, Brazil

ARTICLE INFO

Article history:
Received 5 November 2013
Accepted 22 January 2015
Available online 7 February 2015

Keywords:
Copy-move forgery detection
Interest points
Speed-up Robust Features
Keypoints clustering
Multi-scale analysis
Scale-space
Voting processes
Realistic dataset

ABSTRACT

This work presents a new approach toward copy-move forgery detection based on multi-scale analysis and voting processes of a digital image. Given a suspicious image, we extract interest points robust to scale and rotation finding possible correspondences among them. We cluster correspondent points into regions based on geometric constraints. Thereafter, we construct a multi-scale image representation and for each scale, we examine the generated groups using a descriptor strongly robust to rotation, scaling and partially robust to compression, which decreases the search space of duplicated regions and yields a detection map. The final decision is based on a voting process among all detection maps. We validate the method using various datasets comprising original and realistic image clonings. We compare the proposed method to 15 others from the literature and report promising results.

1. Introduction

In 2004, a research team from Seoul National University led by professor Woo-Suk Hwang became famous publishing in the academic journal *Science* [1], papers showing strong advances in stem cells research. Some months later, however, investigations revealed that some images, which depicted important results of the research, have very likely been tampered with. The scandal and the pressure of the scientific community resulted in paper retractions by *Science* in 2006, and in the professor's dismissal in that same year [2–4].

The above event has direct connection to the technological improvements that our society have been observing. Advanced technology and software have provided users with an active role in the image handling process, from simple naïve adjustments to realistic image manipulations produced with criminal intents.

Copy-move forgery, also known as cloning, is one of the easiest, yet powerful, ways to create fake images. It consists of copying segments of an image and pasting them elsewhere in the same image [5]. The objectives of this tampering are hiding, duplicating or moving elements depicted in the image. Fig. 1, shows a copy-move forgery example.

As only copying and pasting are usually not enough to produce realistic clonings, several additional operations are employed to fill this requirement. For example, if one intends to conceal an element by overlapping it with a texture-like segment (e.g., foliage, sand, etc.), it might be necessary to match the segment with its new neighborhood. That could be accomplished by rotating, resizing, flipping or blurring the copied patch before pasting it. Such transformations also reduce visual traces of tampering. Furthermore, as the cloning is finished, the author could add Gaussian noise or save the image in a lossy compression format like JPEG. This would make the cloning detection even harder to be achieved visually and by computer methods as the copied information is well matched to the background and the compression artifacts make it harder to separate natural telltales resulting from the cloning from the ones resulting from the compression.

Several methods from the literature deal only with simple copy-move forgery scenarios, while other approaches present relevant contributions toward the detection of sophisticated clonings, but still having major limitations. In this sense, we propose a new approach to detect copy-move forgeries in digital images that is focused mainly on investigating and spotting out traces of copy-move forgeries aided by complex operations, such as rotations, resizings and combinations of them. We tackle this problem by using a strategy based on a multi-scale analysis and a voting process. Given a suspicious image, we extract representative interest points robust to scale and rotation transformations from it. Next, we cluster the correspondent points into regions considering

☆ This paper has been recommended for acceptance by Yehoshua Zeevi.

* Corresponding author.
E-mail addresses: ewerton.silva@students.ic.unicamp.br (E. Silva), tjose@ic.unicamp.br (T. Carvalho), anselmoferreira@ic.unicamp.br (A. Ferreira), anderson.rocha@ic.unicamp.br (A. Rocha).

http://dx.doi.org/10.1016/j.jvcir.2015.01.016
1047-3203/

the following geometric constraints: (i) the spatial distance between interest points; (ii) the inclination of the line that links such points relative to the x-axis. We also construct a multi-scale pyramid to represent the image scale-space. Each region (cluster) from each scale is examined as a cloning candidate using a descriptor robust to rotation, scaling and compression. This process decreases the search domain and yields a detection map. The final decision is provided by a voting step among all detection maps, in which a pixel is considered as tampered as long as it is marked so in the majority of the pyramid scales.

To validate the proposed method, we have built a dataset comprising 216 realistic cloning images. Using such dataset, we show qualitative and quantitative experiments and we compare our proposed approach to 15 other methods in the literature. Such dataset is publicly available and was used in the *1st IEEE Intl. Image Forensics Challenge (IFC)*.[1] We also validated the methods on a number of freely available datasets proposed by [6]. All validations considered scenarios divided by operations (simple copy-paste, resizing, rotation, compression) and also combined scenarios with different combinations of operations.

The main contributions of this paper include:

- the development of a new and effective approach, which combines well-founded literature techniques in a novel way, to detect copy-move forgeries able to deal with rotations, resizings, and compression simultaneously;
- the development of a dataset comprising hundreds of realistic copy-move forgeries[2];
- a comparison of the proposed method to 15 others in the literature on different and complex freely available datasets showing their pros and cons on diverse scenarios.

This paper is organized as follows. First, we provide details about the copy-move forgery detection problem and show related work in Section 2. In Section 3, we introduce the new methodology for identifying copy-move forgeries. We present qualitative and quantitative experiments with our method and compare it to 15 others from the literature in Section 4. In Section 5, we conclude the paper and expose some ideas for future work.

2. Related work

The literature has been concerned with copy-move forgery detection in terms of the additional operations often applied to the falsification. Rotation, resizing, horizontal/vertical flipping, edge blurring and white Gaussian noise insertion are the main transformations used to give realism to manipulated images, thus expanding visual and computing efforts to check the image's authenticity. Still, there is the common JPEG compression procedure, which also changes pixel values. The challenge with such operations relies on the fact that, once they are applied, the correspondences between cloned segments become hard to spot out. For instance, a rotation modifies pixel values and positions compared to the original segment.

To deal with this problem, two different schemes are largely adopted in the literature: a block-based comparison and a keypoint-based comparison.

2.1. Block-based comparison schemes

Most of the approaches are included in this category. The main difference among them is the strategy used to describe the blocks of pixels. Fridrich et al. [7] proposed the *Exact Match*, which can be taken as the basis-algorithm for finding duplicated regions. First, the authors employed a square sliding window to collect image blocks, which were gradually stored in a matrix. After all possible blocks have been collected, the matrix was lexicographically sorted, and two consecutive identical lines were deemed as cloned regions. This approach, however, did not deal with additional manipulations (e.g., JPEG compression). To provide some robustness to that, the authors proposed the *Robust Match*: a second approach in which they used the *Discrete Cosine Transform* (DCT) to characterize each block. Using DCT, the method was able to find some duplicated regions under JPEG compression, but it still failed in Gaussian noise, rotation and resizing scenarios.

Popescu and Farid [8] used *Principal Component Analysis* (PCA) to reduce the dimensionality of the blocks and to produce a new representation of them. This increased the robustness of the method under JPEG compression and Gaussian noise addition. Luo et al. [9] extract seven features based on the average of the pixel intensity in each RGB channel, and on some directional information, from each block. The authors reported high detection accuracy rates for JPEG compression, additive Gaussian noise, Gaussian blurring and mixed operations. However, this approach did not tackle rotations, resizings, and flippings.

Mahdian and Saic [10] calculated 24 Blur moment invariants to generate a feature vector representation of each block of pixels. The final feature vector was 72-dimensional, since the calculation of the invariants were performed in each RGB (*red*, *green*, and *blue*) channel separately. Moreover, to reduce the size of this representation, PCA was also employed. By taking this approach, duplicated regions could be effectively pointed out even in the presence of blurring, additive Gaussian noise and JPEG compression of the image. On the other hand, rotation, resizing, and flipping operations were not considered in the experiments.

Zhang et al. [11] proposed an approach based on the analysis of the image's recursive sub-band by using *Discrete Wavelet Transform* (DWT) and the calculus of phase correlations. The performed experiments showed the method's robustness under JPEG compression and blurring, only. Li et al. [12] also used DWT along with *Singular Value Decomposition* (SVD) to decrease the amount of information examined and to generate a more robust block representation. The approach was mainly effective in the cases of JPEG compression with quality factors greater than 70%.

Kang and Wei [13] used SVD to extract feature vectors from overlapping blocks and detect copy-move regions. According to the authors, SVD provides algebraic/geometric invariance, and insensitiveness to noise which is useful in copy-move detection. However, results using rotated and resized cloned parts were not reported.

Ryu et al. [14] proposed to use Zernike moments, which are invariant to rotations to detect duplicated elements. Such work was overhauled later in [15], with a more reliable block matching procedure that incorporates the phase of Zernike moments into a feature space error-reduction procedure, yielding better accuracies. However, resizing was not treated by this method.

Bravo-Solorio and Nandi [16] took the correlation coefficient of the *Fourier Transform* (FT) as the similarity measure between blocks of pixels in log-polar form. They further analyzed and discarded blocks in which the entropy was lower than a threshold. The experiments showed robustness to deal with flipping and simple copy-move operations, but problems when dealing with resizing and rotation.

Bashar et al. [17] proposed an approach where features can be provided by *Discrete Wavelet Transform* (DWT) or by *Kernel Principal Component Analysis* (KPCA). The detection process uses these features individually to characterize blocks of pixels. Features extracted from blocks are organized into a matrix and

[1] http://ifc.recod.ic.unicamp.br/fc.website/index.py.

[2] Upon acceptance of this paper, the will be freely available from http://dx.doi.org/10.6084/m9.figshare.978736.

(a) Original (b) Copy-move forgery

Fig. 1. A simple example of copy-move forgery in which a bird is completely masked.

lexicographically sorted to detect possibly tampered regions. According to the authors, the proposed method cannot handle geometric operations, e.g., resizing and shearing.

Bayram et al. [5] proposed to employ *Fourier-Mellin Transform* (FMT) and *Counting Bloom Filters* (CBF) to describe and sort image blocks, respectively. In spite of increasing efficiency, the use of CBF decreases the effectiveness of the method, since CBF deals only with very similar blocks. The method was able to identify clonings under JPEG compression (quality factors greater than 70%), resizing (up to 5%) and rotation (up to 10°).

Wang et al. [18] proposed the use of a circular sliding window and *Gaussian Pyramid Decomposition* (GPD) to describe image blocks. The main goal was to find correspondences between the rotated blocks and their original counterparts. The method showed good results for rotation, blurring, JPEG compression and horizontal flipping transformations, but was unable to detect duplication under resizing operations.

Lin et al. [19] represented each block by a 9-dimensional feature vector containing information regarding the pixels intensities in some specific regions of the block. Also, the common lexicographically sort process is replaced by the *Radix-Sort* algorithm, which offers a linear time sorting. The algorithm is robust to JPEG compression and Gaussian noise, only. Ardizzone and Mazzola [20] used a *Bit-Plane Analysis* to detect copy-move forgeries. This approach, however, was not robust to post-processing operations.

Wang et al. [21] first applied GPD to the suspicious image, then calculated the Hu moments of each block of pixels. The experiments showed that the approach is robust to Gaussian noise, JPEG compression and blurrings. However, other operations were scarcely contemplated. This includes a 90° rotation, and a horizontal flipping case, only.

Barnes et al. [22] developed a randomized approach to detect similar patches (blocks of pixels) in an image, which is based on top of *PatchMatch Algorithm* [23]. Although the algorithm is not oriented for detecting image forgeries, the authors have used it to detect simple clonings (although they showed only a couple of simple examples in their paper). Also, recently Cozzolino [24] explores PatchMatch algorithm, modifying it to deal with rotation operations.

2.2. Keypoint-based comparison schemes

Looking for a way to solve the copy-move forgery problem without relying solely on block comparison, Huang et al. [25] proposed an approach based on comparisons of *Scale-Invariant Features Transform* (SIFT) [26] descriptors. Such descriptors are reasonably invariant to noise, rotations, resizings and lighting adjustments. The idea was to compare standard SIFT descriptors of previously found keypoints in the image. The method did not work well for small duplicated regions. Pan and Lyu [27,28] also employed SIFT descriptors in a similar fashion, with which the method was able to cope with JPEG compression and Gaussian noises. Moreover, some results showed that the method had potential for finding rotated and resized cloned segments.

Amerini et al. [29] conducted a more complete study on how to use SIFT descriptors to spot out clonings. The analysis of SIFT correspondences was carried out by means of an agglomerative hierarchical clustering procedure. Such step aimed at reducing the false positive rates and providing the method with the ability to detect segments replicated multiple times. Additionally, the clustering is useful to estimate the geometric transformation the cloned segments have undergone. The approach was examined in scenarios comprising a few rotations (5°, 10°, 50°, and 70°), symmetric/asymmetric resizings (with scaling factors up to 2), JPEG compression and other operations. High true positive rates were reported but more diversity in the transfomations and their ranges should be considered.

Alternatively to SIFT descriptors, Xu et al. [30] used the *Speeded-Up Robust Features* (SURF) [31] algorithm to locate copy-move regions. The authors compared SURF descriptors from an image by randomly dividing the SURF keypoints set into two groups, finding the nearest neighbors in such groups, saving the matchings and repeating the process for each sub-group until all of them become unitary. The authors reported experiments with rotation, resizing and blurring segments along with white Gaussian noise insertion. Combinations of these operations were also considered. However, such method may have difficulties in locating cloned homogeneous segments, as they could not provide enough keypoints for a proper analysis.

Shivakumar and Baboo [32] proposed a methodology in which SURF keypoints are extracted and stored in a Kd-tree. This data structure is used to improve and facilitate the search for keypoint matchings (nearest neighbors), and, according to the authors, to generate lower false negative rates. The paper lacks a deep experimental validation, though, by only presenting visual results of the detection in two cases of rotation, resizing, and Gaussian noise independently.

Christlein et al. [6] investigated several copy-move forgery approaches [10,21,14,8,13,17,9,16,19,18,7,5,25,27,29,30,32], and performed a detailed analysis of their performance under different scenarios. The experiments showed that the approaches [18,14,16] yielded good results for rotation, along with the keypoint-based strategies. For scaling operations, it was reported that by using Zernike moments, KPCA, PCA, or DCT, it might be possible to achieve satisfactory detection results. On the other hand, the resilience to JPEG artifacts were most seen in the keypoint-based methods, and also when applying PCA or DCT. Finally, under those post-processing operations combined, the keypoint-based approaches reached the best precision results in the most critical scenarios, i.e., when large rotations, resizings and high compressions are performed on the cloned segments, whereas the effectiveness of the remaining approaches decrease significantly.

When employing image descriptors [25,27,30,32], reported few detection results (most of them being visual) under combined operations, whereas in [29], the authors contemplate several combined operations for SIFT features. Given that this is an important scenario in which copy-move forgery detection algorithms must improve, in the next section, we present the proposed methodology to deal with copy-move forgeries containing additional transformations and their combinations.

3. Methodology

Inspired by some literature approaches that aim at dealing with rotation, resizing and JPEG compression operations, we have developed a novel approach that involves keypoint matching, multi-scale analysis and voting of cloned candidate regions. Our method comprises seven steps: (i) image pre-processing, (ii) keypoints detection, (iii) keypoints matching, (iv) clustering, (v) pyramidal decomposition, (vi) multi-scale analysis, and, finally, (vii) voting. The general scheme of our method is depicted in Fig. 2.

3.1. Step I – Image pre-processing

The first step of our strategy consists of converting the color space of the input image from RGB (Red, Green and Blue) to HSV (Hue, Saturation and Value). By doing this, we aim at decreasing the amount of false positives that are frequently found on homogeneous regions of the scene, such as blue sky, untextured walls, and strongly bright or dark regions. Fig. 3 depicts an example of how the number of false positives is decreased by using HSV color space conversion instead of simply using RGB input images.

When evaluating regions on the HSV domain, it is possible to highlight the difference between naturally similar regions which are not really cloned, preventing the occurrence of some mismatches. This can be justified by the fact that, in opposition to the RGB space, the intensity information (V) is detached from the color (HS) information in the HSV domain. Therefore, two very similar regions wrongly classified as cloned in the RGB space can be classified correctly in a color space where intensity values are taken independently from color values. From a semantic point of view, it means that color and intensity information, when individually evaluated, may amplify and highlight differences between two resembling regions.

We have used the OpenCV [33] library to convert from RGB to HSV. However, the conversion method generated Hue values in the range of $[0, 179]$, whereas the other channels (SV) showed dynamic ranges of $[0, 255]$. This fact could favor the S and V channels during the calculation of distances. Since all channels should influence equally to the final distance value, we normalize all H, S and V values of a pixel block according to

$$p'_C = \frac{p_C - \min(\mathrm{HSV})}{\max(\mathrm{HSV}) - \min(\mathrm{HSV})}, \quad (1)$$

where p_C and p'_C are values from a channel C ($C \in \{H, S, V\}$) in a specific pixel p before and after the normalization, respectively, and min(HSV) and max(HSV) represent the minimum and maximum values among all channels and from all pixels in the examined block.

3.2. Step II – Keypoints detection

As previously discussed, detecting duplicated blocks under some transformations is a paramount challenge in copy-move forgery detection. To make our approach invariant to scale (resizing) and rotation operations, in this step, we extract interest points

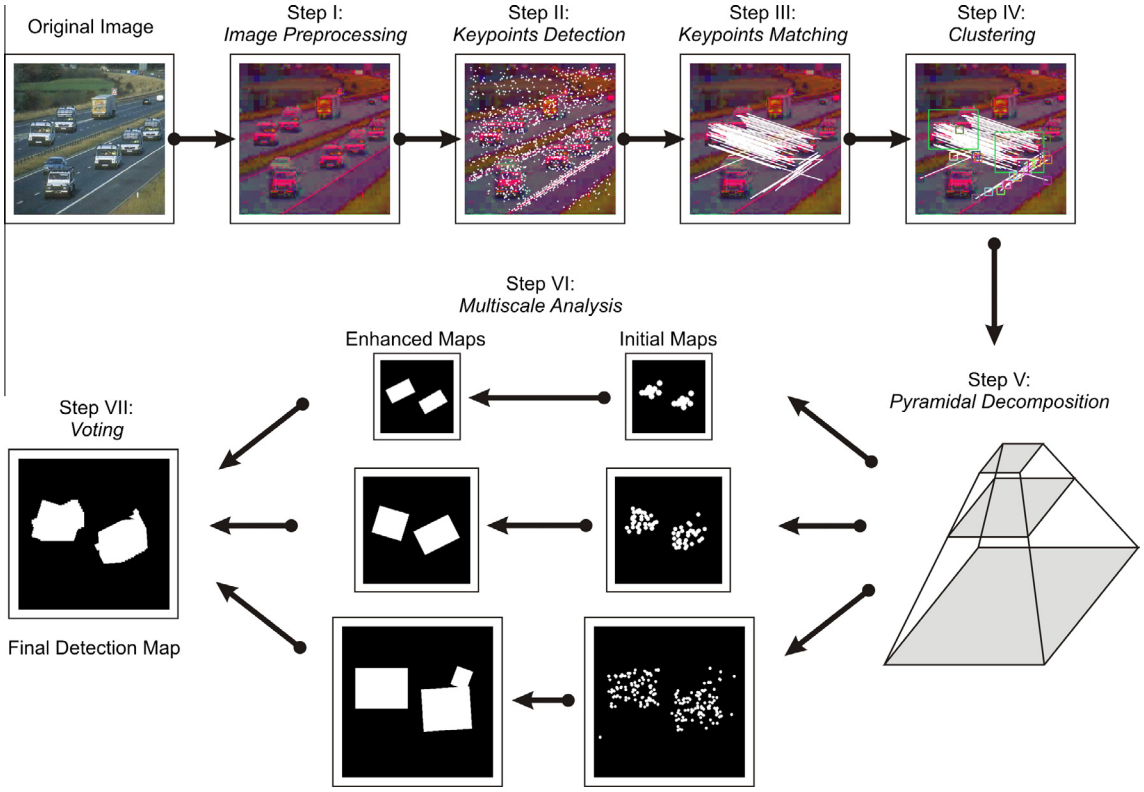
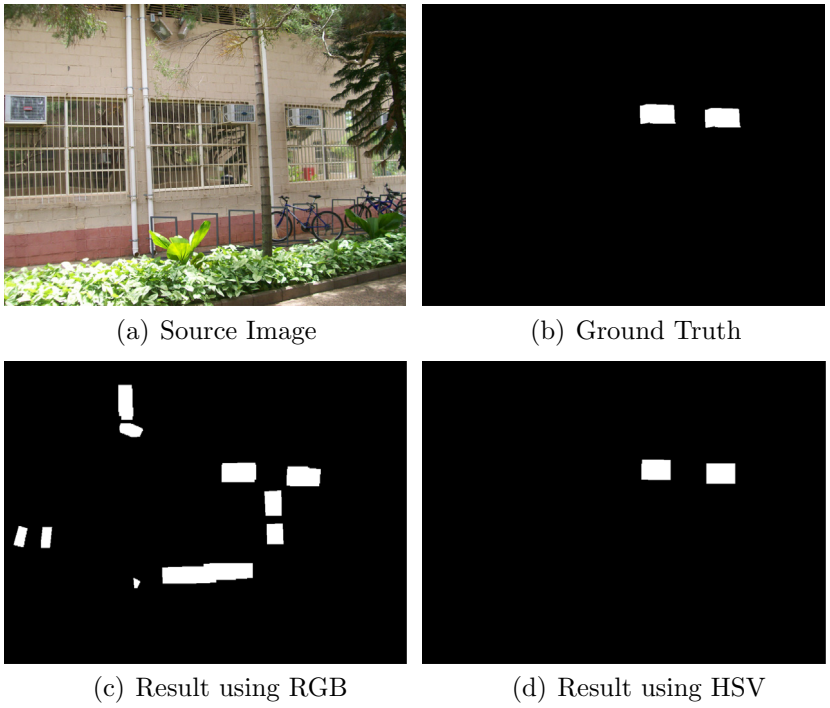


Fig. 2. General scheme of our approach. The input image is subject to several examinations. First, the color space conversion (RGB to HSV) takes place. Then, representative keypoints and their descriptors are localized in the image. Next, these keypoints are matched and clustered. The image's scale-space is also generated by a Gaussian pyramidal decomposition. Thereafter, a multiscale analysis is employed to investigate the candidate cloned regions in each scale of the pyramid. This process generates partial detection maps that are used to determine the final duplicated regions by a voting-based strategy. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Source Image (b) Ground Truth

(c) Result using RGB (d) Result using HSV

Fig. 3. Binary maps produced by the proposed method with and without HSV conversion. The number of false positives is significantly increased when not using HSV color space conversion for the reasons explained in the text. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(a) Keypoints extracted using SURF detector (Step II) (b) Keypoints matching (Step III)

Fig. 4. Examples of generated results after steps II and III.

(keypoints) from the image, by using the *Speeded-Up Robust Features* (SURF) algorithm [31]. Fig. 4(a) shows an example of keypoints detection performed by the SURF algorithm.

To detect keypoints, SURF uses an approximation to the Hessian matrix along with box filters, integral images, and non-maximum suppression. On the descriptor side, given a set of keypoints, each one is described by using Haar Wavelet responses on vertical and horizontal directions from a circular region around them. The response values are weighted by a Gaussian window and compose the final 64-d descriptor for a specific keypoint.

Given that the SURF algorithm runs over grayscale images, we need to convert the HSV image obtained in Step I accordingly (or without losing generality, we can use the *V* channel directly). After that, we just run SURF on the image to extract its keypoints and their descriptors. We save all the information collected and proceed to the next steps considering the color HSV image (the grayscale version can be discarded).

3.3. Step III – Keypoints matching

After collecting the image keypoints and descriptors, we calculate correspondences among them. We define that a keypoint should have only one matching throughout the image. To ensure this, we use the *Nearest Neighbor Distance Ratio* (NNDR) policy [34].

The classical Nearest Neighbor (NN) heuristic performs a matching between two keypoints *A* and *B*, only when the distance between their descriptors is lower than a similarity distance threshold. When multiple matchings are found, only the one with the smallest distance will be taken. The drawback of employing NN is that it may be hard to determine a good threshold value, since the image and the cloned segments might have undergone several transformations (e.g., rotations, flipping) [35,36].

A feasible alternative is the NNDR heuristic policy, which takes the ratio between the best and the second best neighbor (of a

specific point) to establish a correspondence. Given a keypoint A, its nearest neighbor B is determined by also bringing its second nearest neighbor C to the table. A and B will be paired only if the distance between them is above a minimum similarity distance threshold, and if this distance is smaller than the distance between A and C multiplied by a predefined *Score*:

$$Score * D_m(A,C) \geqslant D_m(A,B), \quad (2)$$

where D_m represents the distance metric between two keypoints' descriptors.

The chosen *Score* has to be greater than or equal to the distance ratio, in order to match A and B. The idea of NNDR is that a true correspondence to A has to be very close (similar) to it compared to the closest false matching to A, thereby punishing keypoints with high similarity to several others in the image.

Our approach uses the Euclidean Distance between descriptors to evaluate similarities. To decrease the number of false positives, we apply another threshold over the spatial distances, preventing close keypoints to be incorrectly paired. Fig. 4(b) depicts an example of matching.

3.4. Step IV – Clustering

This step decreases the search space for copy-move regions. Also, by delimiting the regions to be compared, the clustering avoids the detection of some false positives.

A clustering yields two matching groups: source and destination, which will be later examined separately. They are considered as correspondent regions inside the image and are good cloning candidates. To obtain these regions, the clustering approach checks the following constraints:

- **Spatial proximity between keypoints from the same group:** consider that we have a match between keypoints A and B. They might integrate a general group G, for instance. In this situation, A might belong to G_{source} subgroup, and B might belong to $G_{destination}$ subgroup, or vice versa. For a subgroup to admit a paired keypoint (e.g., A) as a new member, the spatial distance between the keypoint and its nearest keypoint (also in spatial terms) in such subgroup needs to be smaller than a predefined threshold. Also, it is necessary to analyze both matched keypoints, since they have to be in the same general group, but in different subgroups (source and destination);
- **Correspondence angle:** the angle, relative to the horizontal axis, between the line that links two matched keypoints is an important constraint. Given two correspondent subgroups (e.g., G_{source} and $G_{destination}$), a new keypoint A candidate to be included into G_{source} will be effectively included into G_{source}, only if the angle of the line that connects the candidate point A and its matching keypoint B is within the same range of the other points in G_{source}. Using four range partitions ($[0^\circ, 89^\circ]$, $[90^\circ, 179^\circ]$, $[180^\circ, 269^\circ]$ and $[270^\circ, 359^\circ]$), for instance, the angle of the line that connects A and B needs to integrate the same range of the other keypoints from G_{source}.

Fig. 5 illustrates examples of these constraints.

Correspondent regions (subgroups) are delimited by the spatial distribution of their keypoints and we use rectangular windows to bound them. The width of a window is taken as the larger horizontal distance between two keypoints from the subgroup, plus a previously defined value (the height calculus is analogous). The goal of such increment is gathering segments that could be part of the duplication, but in which the SURF algorithm found no interest points. Algorithm 1 summarizes the entire clustering process.

Algorithm 1. Clustering Process.

Input: M (a set comprising all the keypoints matchings produced by Step III. Each keypoint matching m is composed by a source point A and a destination point B)
n (the number of partition ranges) *threshold* (highest spatial distance between keypoints)
Output: P

```
Construct P = {p_1, p_2, ..., p_n} where p_i is an empty partition
(e.g., for n = 4, p_1 is a partition comprising angles between
[0°, 89°]; p_2 comprises angles between [90°, 179°], etc.);
for each matching m ∈ M do
  Calculate the correspondence angle c relative to m;
  Find partition p_i where c ∈ p_i;
  if p_i == ∅ then
    create a new group G (G = G_source ∪ G_destination) associated
with p_i;
    G_source ← A;
    G_destination ← B;
  else
    F = false;
      for each group G ∈ p_i do
        for each point X ∈ G_source do
          d_A = Euclidean distance between A and X;
          d_B = Euclidean distance between B and Y;
          (Y ∈ G_destination and Y matches X)
          if d_A < threshold and d_B < threshold then
            F = true;
          end if
        end for
        if F == true then
          G_source ← A;
          G_destination ← B;
        end if
      endfor
      if F == false then
          create a new group G (G = G_source ∪ G_destination) ∈ p_i;
          G_source ← A;
          G_destination ← B;
        end if
      end if
    end for
  return P;
```

3.5. Step V – Pyramidal decomposition

In this step, we perform a Gaussian Pyramidal Decomposition of the image. The process first uses a Gaussian filter for blurring the image. Then, even rows and columns are removed, which generates an image with size 25% of the original image size. The next pyramid level is obtained by repeating those steps to the new smaller image, and so further [37]. In the next step, all images in the pyramid are individually analyzed.

3.6. Step VI – Multi-scale analysis

This step represents one of our main contributions. Here, we perform the analysis of the input image by examining its various scales in a Gaussian pyramidal decomposition. The process is somewhat similar to other previously literature approaches (e.g., Robust Matching [7]): a window of constant size slides through the image collecting pixel blocks or descriptors; these are linearized and then stored in a matrix; this data structure is lexicographically sorted and the search for duplicated lines starts. A

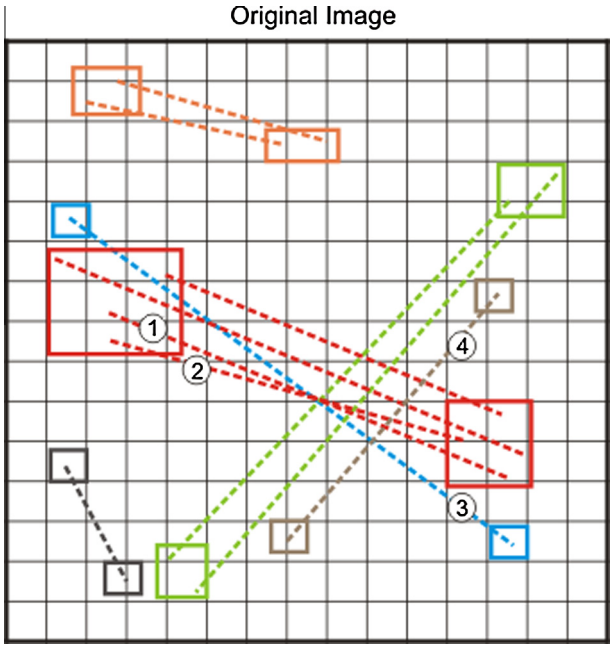


Fig. 5. Clustering example. Consider that the correspondence angle of lines 1 and 2 are in the same orientation range. Given that there is also spatial proximity between their keypoints, they could be part of the red group. In opposition, the blue keypoints (line 3) could not integrate the red group, since their correspondence angle is not in the required range for such group. Finally, the correspondence angle of line 4 (brown) could help their keypoints to be included to the green group (angles are very close to each other), but the spatial proximity constraint does not allow that. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

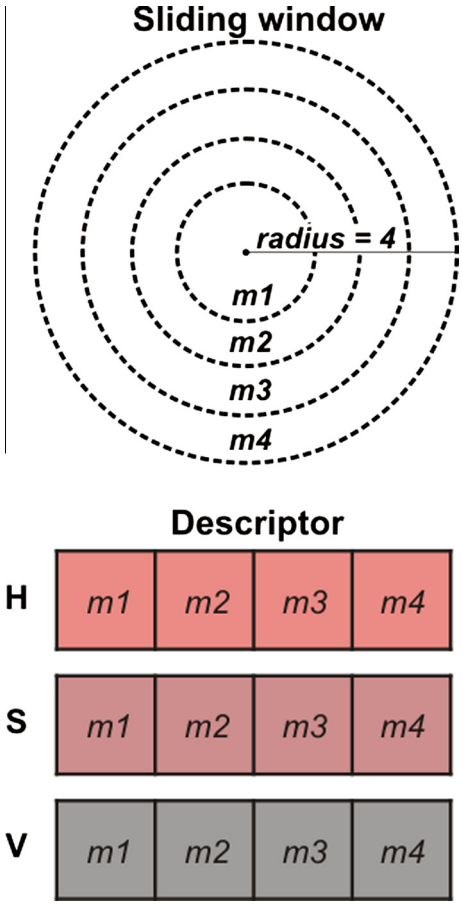


Fig. 6. Sliding window and descriptor used to collect and represent pixel blocks.

binary detection map is generated for the comparison of the obtained result to a reference map (the groundtruth). Although similar to [7], our approach presents fundamental differences:

1. We perform a group resizing procedure in all of the images provided by the pyramidal decomposition, with the exception to the one at the lowest level. The procedure takes place after the analysis of the original image is finished. We downscale the subgroups (from the original image) proportionally to the image's decreasing size factor. For instance, from the first level of the pyramid to the second, the original image will have its width and height decreased by a factor of $\frac{1}{2}$. Such factor is used to adjust the groups accordingly, thus preserving their semantic content through the scales. Analogously, the resizing factor is $\frac{1}{4}$ for the next level, and so on. This step aims at ensuring that the inspected image regions (groups) in upper pyramid levels will contain equivalent information compared to the corresponding regions in the original image.
2. We use a circular window and a descriptor analogous to the one proposed by Wang et al. [18]. We collect the mean of the four concentric circles inside the window for each HSV channel, which totalizes 12 dimensions. Fig. 6 illustrates the descriptor values.
3. We do not slide the window through the entire image, but rather through the correspondent regions (groups) found by the clustering step.
4. Two matching regions (source and destination subgroups) are analyzed independently, preventing possible outliers from disturbing the detection process. This procedure decreases the false-positive rate and increases accuracy.
5. Before the descriptor extraction, a pixel block must be normalized according to Step I.
6. After the lexicographical sorting, two blocks can be taken as duplicated only if they belong to distinct subgroups (source and destination).
7. The analysis of each image in the pyramid generates a new Partial Detection Map (PDM), which is a detection map for the image at a given pyramid level.
8. We perform a region gathering procedure in each PDM. We join cloning candidate regions from a subgroup by merging segments in a unique region using a *Minimum Bounding Rectangle* (MBR) algorithm. Therefore, a larger area of a subgroup becomes a potentially cloned region. For instance, the fifth PDM refers to the image in the fifth pyramid level. The analysis of this partial map is carried out only within the subgroups of that image. The MBR algorithm is then applied to each subgroup (source and destination) independently, and not over all the good cloning candidates from a group (in this case, source and destination subgroups would be treated as a unique region, and thus would be allocated in a single minimum bounding rectangle, which is not the case). A detailed overview of this step is depicted in Fig. 7.

3.7. Step VII – Voting

When looking for cloned regions in an image, we usually face a challenging problem: if we increase the similarity distance threshold in order to effectively find cloned regions, the number of incorrect detections may also increase. As a good solution to attenuate this problem, we propose combining the multi-scale analysis to a voting process.

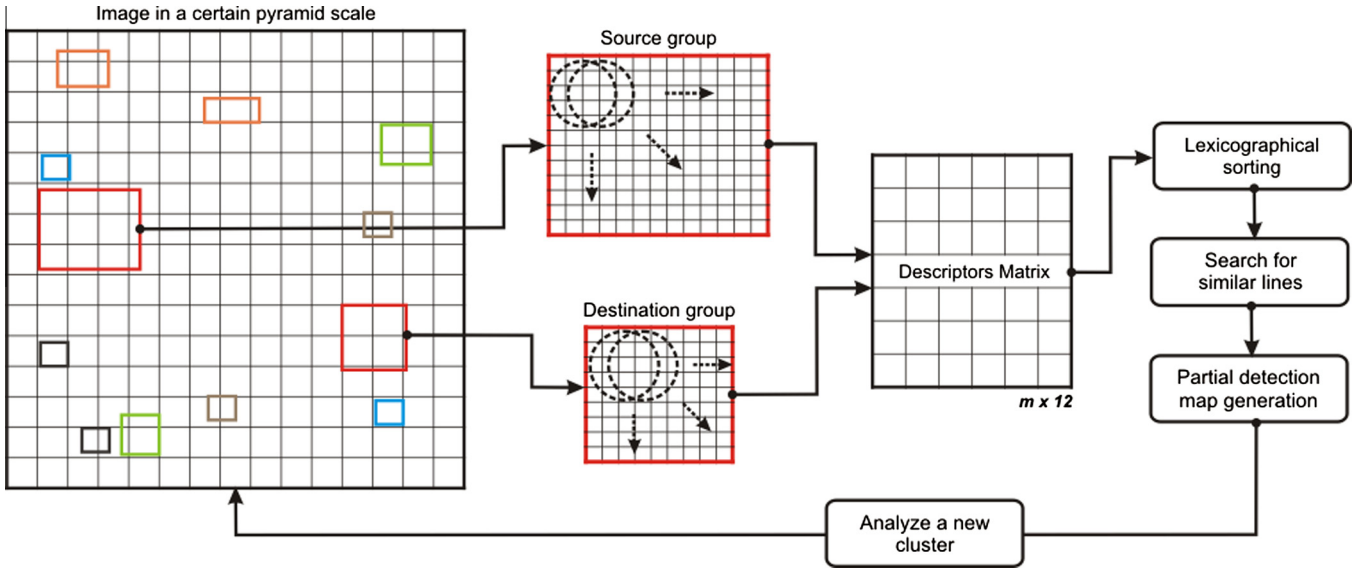


Fig. 7. Step VI overview. In a given pyramid level, the red group is analyzed (source and destination) and its blocks descriptors are stored in a matrix to be lexicographically sorted. Then, a search for similar matrix lines is performed, which yields a partial detection map for the red group. These steps are repeated for all groups, with several updatings of the partial detection map. When all clusters have been examined, the final detection map (for that specific image) is generated. This whole process is applied separately to all scales in the pyramid. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

This step takes the (final) detection maps from all images in the Gaussian pyramid. It considers as duplicated segments only those classified as such in the majority of the pyramid levels. Since the detection maps have different sizes, we simply resize the small ones to match the size of the larger image. Such procedure allows the elimination of some false positives found, mainly, in the original image, and the shape enhancement of the detected regions.

4. Experiments and validation

This section describes the performed experiments to validate the proposed copy-move forgery detection approach. We organized the experiments in three rounds:

- **Round #1:** In this round of experiments, we focused on comparing the performance of the proposed method against 15 state-of-the-art methods, including recent literature methods that deal with complex operations used in copy-move manipulations. Our method is compared to the others on a proposed database of ours referred to as *Copy-Move Hard (CMH)* because it comprises hard-to-detect operations. The dataset is publicly available[3] and some of its images were also used in the *1st IEEE Intl. Image Forensics Challenge (IFC)*, 2013.[4] In this round of experiments, besides testing all methods on the full dataset (with and without compression), we evaluated the methods for different subsets: one subset with simpler copy-and-paste operations, one comprising rotation operations, one containing resizing, one for rotation and resizing altogether, and finally, one containing all operations combined (rotation, resizing, and compression).
- **Round #2:** In this round of experiments, we focused on showing the performance of the proposed method in different and complementary datasets. For this purpose, we considered datasets published by Christlein et al. [6] henceforward referred to as *Copy-Move Erlangen-Nuremberg* (CMEN) comprising easy and hard-to-detect operations generated by the Erlangen-Nuremberg research group led by [6]. To compare the performance of the proposed method against state-of-the-art methods, for each subset in the previous round of experiments, the three best state-of-the-art methods have been selected and tested. We evaluated the top performers in order not to overload the reader with the results of methods that in fact are not better than the ones selected.
- **Round #3:** In this last round of experiments, we performed a qualitative analysis of the proposed method. We show visual results (binary detection maps) for some images from CMH comprising different operations and challenging tampering.

4.1. Datasets

In this paper, we considered the datasets described below for performing the experiments.

4.1.1. CMH dataset

The first dataset used in the experiments was created by our research group and comprises 108 realistic cloning images. Each image is stored in the PNG format (which does not modify pixel values), and has a resolution varying from 845×634 pixels (the smallest) to 1296×972 pixels (the biggest). The dataset consists of four parts:

- CMH_{P1}: 23 images where the cloned area was only copied and then moved (simple case);
- CMH_{P2}: 25 images with a rotation of the duplicated area (orientations in the range of $-90°$ and $180°$);
- CMH_{P3}: 25 images with a resizing of the cloned area (scaling factors between 80% and 154%);
- CMH_{P4}: 35 images involving rotation and resizing altogether.

Every image has its own binary reference map (ground truth) indicating the original and cloned regions in white color.

Additionally, to address compressions, we have compressed the full dataset ($CMH_{P1} \cup CMH_{P2} \cup CMH_{P3} \cup CMH_{P4}$) in JPEG format using quality factors of 70%, 80%, and 90%. To avoid favoring any specific quality factor, from the entire dataset, we randomly select 36 images, and compress them with a quality factor of 70%. Next, from the remaining 72 images, we randomly select 36 images,

[3] http://dx.doi.org/10.6084/m9.figshare.978736.
[4] http://ifc.recod.ic.unicamp.br/fc.website/index.py.

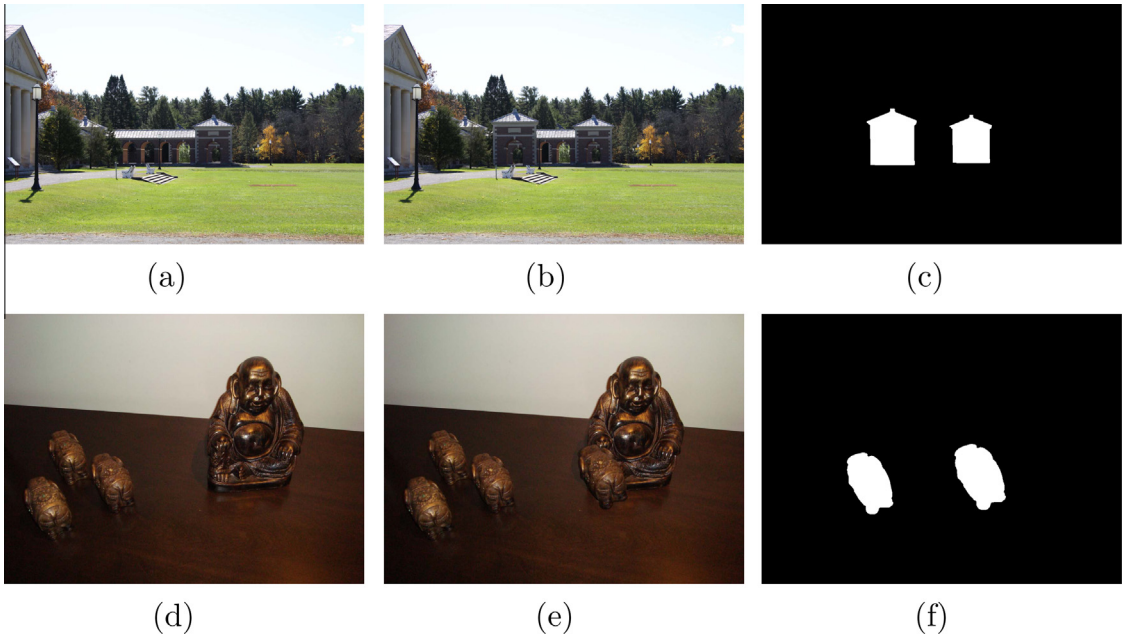


Fig. 8. CMH dataset examples. (a) and (d) are the original images; (b) and (e) are cloning examples, and (c) and (f) are their reference maps.

and compress them with a quality factor of 80%. Finally, we compress the remaining 36 images using a quality factor of 90%. Therefore, the final dataset is composed of 216 images, half of them being uncompressed. Fig. 8 depicts some examples of images from the dataset. All images can be downloaded at: http://dx.doi.org/10.6084/m9.figshare.978736. Upon paper acceptance, the source code of all the introduced algorithms will also be freely available (via GitHub).

For more details about the dataset setup, please refer to Appendix A.

4.1.2. CMEN datasets

The CMEN datasets refer to a subset of benchmarking datasets created by Christlein et al. [6] when comparing several copy-move detection methods. It comprises 336 images stored in PNG format with a resolution varying from 800×533 pixels (the smallest) to 3872×2592 pixels (the biggest). CMEN datasets comprise:

- $CMEN_{P1}$**:** 48 images where the cloned area was only copied and then moved (simple case);
- $CMEN_{P2}$**:** 144 images with a rotation of the duplicated area (orientations of 2°, 4°, 6°, 8°, 10°, 20°, 60° and 180°);
- $CMEN_{P3}$**:** 144 images with a resizing of the cloned area (scaling factors of 50%, 80%, 91%, 93%, 95%, 97%, 99%, 101%, 103%, 105%, 107%, 109%, 120%, 200%).

It is worth noting that Christlein et al. [6] did not provide images in which rotation and scale operations have been applied together. As for the CMH dataset, every image has its own binary reference map (ground truth) indicating the original and cloned regions in white color.

4.2. Evaluation metrics

For evaluating all the methods, we chose the three following metrics, in accordance to the ones used in the *1st IEEE Intl. Image Forensics Challenge in 2013 (IFC):*

- **True Positive Rate (*TPR*):** indicates the percentage of correctly located cloned regions. It is calculated as

$$TPR = \frac{|TP|}{|R_{clone}|}, \tag{3}$$

where $|TP|$ (True Positive) represents the number of pixels correctly classified as cloned in the detection map, and $|R_{clone}|$ represents the number of real cloned pixels in the reference map.

- **False Positive Rate (*FPR*):** indicates the percentage of incorrectly located cloned regions. It is calculated as

$$FPR = \frac{|FP|}{|R_{normal}|}, \tag{4}$$

where $|FP|$ (False Positive) represents the number of pixels wrongly classified as cloned in the detection map, and $|R_{normal}|$ represents the number of pixels, in the reference map, that do not belong to the cloning regions.

- **Accuracy (*ACC*):** gives the quality of detection based on *TPR* and *TNR* (True Negative Rate), which indicates the percentage of correctly located non-cloned regions. It is calculated as

$$ACC = \frac{TPR + (1 - FPR)}{2}, \tag{5}$$

Table 1
Label associated with each state-of-the-art method.

Method	Label
Mahdian and Saic [10]	BLUR
Wang et al. [18]	CIRCLE
Fridrich et al. [7]	DCT
Bashar et al. [17]	DWT
Bayram et al. [5]	FMT
Wang et al. [21]	HU
Lin et al. [19]	LIN
Bashar et al. [17]	KPCA
Popescu and Farid [8]	PCA
Shivakumar and Baboo [32]	SIFT
Shivakumar and Baboo [32]	SURF
Amerini et al. [29]	HIERARCH-SIFT
Kang and Wei [13]	SVD
Ryu et al. [14]	ZERNIKE
Ryu et al. [15]	ZERNIKE2

Table 2
Comparative results between state-of-the-art methods and our method for CMH_{P1}, simple clonings.

Method	Avg. ACC	Std. ACC (%)	Avg. TPR (%)	Std. TPR (%)	Avg. FPR (%)	Std. FPR (%)
Proposed Method	**96.19**	**2.54**	**94.08**	**5.38**	**1.70**	**1.73**
SURF	94.41	14.04	89.05	28.15	0.22	0.26
ZERNIKE2	90.10	17.33	83.74	29.53	0.03	0.07
DWT	89.22	3.15	78.46	6.32	0.03	0.04
CIRCLE	88.83	3.16	77.72	6.27	0.05	0.19
ZERNIKE	88.08	3.25	76.24	6.61	0.08	0.37
SVD	87.53	3.11	75.10	6.24	0.04	0.10
KPCA	87.44	3.53	75.17	7.33	0.29	1.37
DCT	86.00	3.71	72.02	7.42	0.02	0.07
FMT	85.35	3.59	71.06	7.36	0.36	1.74
LIN	85.13	3.88	70.31	7.82	0.05	0.24
HU	84.42	3.97	68.97	7.96	0.13	0.38
BLUR	84.09	4.09	68.22	8.21	0.04	0.18
PCA	83.47	4.16	66.93	8.33	0.00	0.00
SIFT	81.56	23.61	63.25	47.33	0.14	0.18
HIERARCH-SIFT	73.95	21.99	48.11	44.18	0.20	0.27

Table 3
Comparative results between state-of-the-art methods and the proposed approach for CMH_{P2}, clonings comprising rotation operations.

Method	Avg. ACC (%)	Std. ACC (%)	Avg. TPR (%)	Std. TPR (%)	Avg. FPR (%)	Std. FPR (%)
Proposed Method	**84.56**	**13.40**	**69.77**	**27.13**	**0.66**	**0.56**
SURF	82.99	21.41	66.11	42.91	0.14	0.17
SIFT	79.68	23.51	59.49	47.13	0.13	0.16
HIERARCH-SIFT	73.36	22.74	46.88	45.62	0.14	0.20
ZERNIKE	64.42	12.91	28.85	25.83	0.02	0.05
CIRCLE	59.03	9.16	18.08	18.32	0.01	0.05
ZERNIKE2	59.02	20.85	24.25	38.60	6.20	8.16
KPCA	57.20	11.21	14.41	22.44	0.01	0.05
DCT	57.09	11.25	14.19	22.51	0.00	0.00
DWT	57.03	11.15	14.08	22.31	0.02	0.04
HU	52.91	7.66	5.86	15.36	0.04	0.08
FMT	52.88	8.11	5.77	16.21	0.01	0.03
SVD	50.76	1.92	1.53	3.83	0.01	0.03
LIN	50.61	2.45	1.23	4.89	0.00	0.00
BLUR	50.30	0.89	0.61	1.78	0.00	0.00
PCA	50.00	0.02	0.00	0.00	0.01	0.04

where $(1 - FPR)$ represents the *TNR*.

In the experiments, we adopt a pixel-level approach to evaluate the detection maps. This approach has been chosen mainly because it is more accurate to detect cloned regions, because it was preferred metric used in the *IFC* 2013, and because in a forensic scenario, experts usually need to point out potential tampered areas to further judgements. It is worth mentioning that recent trends in the Information Forensics Community have pushed for pixel-level forgery detection and localization instead of only image-wise binary metrics.

4.3. Selection of parameters and discussion

To perform the experiments, we have tested our method with the configuration of parameters below. Both spatial and similarity distances were calculated as the Euclidean norm.

4.3.1. Clustering

- highest spatial distance between keypoints inside a subgroup: **50 pixels** (used to decide whether a point will integrate a subgroup);
- lowest spatial distance between matched keypoints: **50 pixels** (if lower, points will not proceed to the NNDR policy analysis);
- score for the NNDR policy: **0.6**;
- group size increment: **10 pixels** in each direction (up, right, bottom and left), when possible;
- number of orientation ranges: **20 ranges** ($[0^\circ, 17^\circ]$, $[18^\circ, 35^\circ]$, $[36^\circ, 53^\circ]$, etc.).

4.3.2. Multi-scale analysis

- sliding window size: **9 × 9 pixels** (radius equal to four pixels);
- number of decomposition steps: **2 steps** (in this case, the pyramid has 3 levels, the first being represented by the original image);
- minimum number of keypoint correspondences in a group: **3** (if lower, the group under investigation is discarded);
- number of neighboring lines to be compared in the sorted descriptor's matrix: **5 lines** (given a block description in the matrix, the 5 subsequent block descriptions will be compared to it);
- minimum spatial distance between pixel blocks: **30 pixels** (if lower, the group under investigation is discarded);
- minimum similarity distance between descriptors: **0.05** (after checking the spatial distance between two blocks with the above threshold, their descriptions are compared);
- minimum number of detections in a group: **3 matchings** (if a group has less than 3 duplicated blocks, it is discarded).

The aforementioned parameter selection was performed by a visual analysis with a small set of images that we have created. This additional dataset included JPEG compression, rotations, resizings and combinations of these operations. Images from such small dataset were not used in the experiments whatsoever.

Table 4
Comparative results between state-of-the-art methods and the proposed approach for CMH_{P3}, clonings comprising resizing/scaling operations.

Method	Avg. ACC (%)	Std. ACC (%)	Avg. TPR (%)	Std. TPR (%)	Avg. FPR (%)	Std. FPR (%)
Proposed Method	**81.56**	**13.68**	**64.10**	**28.29**	**1.24**	**2.46**
SIFT	76.90	23.83	54.03	47.59	0.23	0.64
SURF	76.54	21.73	53.23	43.47	0.14	0.19
HIERARCH-SIFT	73.98	21.66	48.14	43.49	0.16	0.20
DWT	65.14	15.38	30.30	30.77	0.02	0.04
KPCA	64.34	15.78	29.11	31.17	0.42	1.51
ZERNIKE	64.08	13.54	28.25	27.02	0.09	0.39
DCT	62.65	14.42	25.35	28.78	0.05	0.19
ZERNIKE2	56.07	18.50	17.82	34.26	5.67	5.23
FMT	52.48	6.37	5.00	12.73	0.03	0.13
CIRCLE	51.42	2.91	2.86	5.82	0.03	0.08
LIN	50.77	2.31	1.60	4.60	0.05	0.23
SVD	50.36	1.12	0.75	2.23	0.03	0.10
HU	50.13	0.56	0.33	1.10	0.08	0.27
BLUR	50.07	0.37	0.20	0.69	0.07	0.22
PCA	49.97	0.11	0.00	0.00	0.05	0.22

Table 5
Comparative results between state-of-the-art methods and the proposed approach for CMH_{P4}, clonings comprising resizing/scaling and rotation operations.

Method	Avg. ACC (%)	Std. ACC (%)	Avg. TPR (%)	Std. TPR (%)	Avg. FPR (%)	Std. FPR (%)
Proposed Method	**81.60**	**13.20**	**64.48**	**26.42**	**1.29**	**2.18**
SURF	78.18	24.46	59.53	41.74	0.23	0.25
HIERARCH-SIFT	78.09	21.14	56.43	42.50	0.24	0.27
SIFT	72.33	26.67	47.96	47.70	0.36	1.01
ZERNIKE2	57.58	20.36	20.88	37.58	5.71	9.05
ZERNIKE	55.84	15.03	14.98	22.54	0.36	1.67
DWT	54.79	14.70	12.84	21.99	0.32	1.68
KPCA	52.68	13.09	8.32	18.47	0.01	0.07
DCT	52.58	12.44	8.16	16.63	0.05	0.12
CIRCLE	51.44	10.23	6.02	9.26	0.20	1.07
HU	49.18	8.82	1.37	3.08	0.07	0.19
FMT	48.83	8.69	0.63	2.10	0.03	0.14
SVD	48.81	8.74	0.60	2.88	0.03	0.14
LIN	48.71	8.63	0.36	1.22	0.00	0.00
BLUR	48.58	8.59	0.11	0.40	0.01	0.03
PCA	48.55	8.58	0.10	0.58	0.07	0.21

Table 6
Comparative results between state-of-the-art methods and the proposed approach for full dataset ($CMH_{P1} \cup CMH_{P2} \cup CMH_{P3} \cup CMH_{P4}$).

Method	Avg. ACC (%)	Std. ACC (%)	Avg. TPR (%)	Std. TPR (%)	Avg. FPR (%)	Std. FPR (%)
Proposed Method	**85.35**	**13.14**	**71.92**	**26.69**	**1.22**	**1.92**
SURF	82.36	21.98	65.82	41.50	0.19	0.22
SIFT	77.11	24.57	55.35	47.17	0.23	0.66
HIERARCH-SIFT	75.13	21.61	50.45	43.40	0.19	0.25
ZERNIKE	65.68	17.05	34.43	31.60	0.15	0.97
ZERNIKE2	64.47	23.39	34.31	43.54	5.35	7.72
DWT	65.13	18.05	31.30	33.85	0.12	0.94
KPCA	63.94	17.65	28.97	33.17	0.17	0.98
DCT	63.17	16.85	27.29	31.48	0.03	0.12
CIRCLE	61.15	16.55	23.32	30.94	0.09	0.61
FMT	58.42	15.84	17.87	29.77	0.09	0.81
SVD	57.88	16.35	16.72	30.76	0.03	0.10
HU	57.77	15.37	16.55	28.70	0.08	0.25
LIN	57.40	15.47	15.76	28.92	0.02	0.16
BLUR	56.90	15.13	14.75	28.21	0.03	0.14
PCA	56.66	14.93	14.29	27.77	0.03	0.16

Score, *number of orientation ranges*, *minimum similarity distance*, and *minimum number of correspondences in a group* are the parameters that influence the detection most. The first one manages the number of keypoint matchings found during Step III of our method. We noticed that score values below 0.6 generate several correspondences between unrelated keypoints, whereas higher values usually suppress important matchings. For the orientation ranges, we have found 20 to be a good value. For lower values, the clustering step generates larger groups containing several non-similar regions, while higher values increase the number of small groups, thereby decreasing the *TPR*. The minimum similarity distance has been defined as 0.05, since it has generated the best visual detection results during the pre-testing phase with another unrelated dataset as explained above. Finally, constraining the inspection only to groups with a minimum of three keypoint correspondences helps decreasing the number of regions falsely evidenced as image tampering, since an image might depict several naturally similar contents.

Table 7
Comparative results between state-of-the-art methods and the proposed approach for full dataset ($CMH_{P1} \cup CMH_{P2} \cup CMH_{P3} \cup CMH_{P4}$) containing JPEG compression.

Method	Avg. ACC (%)	Std. ACC (%)	Avg. TPR (%)	Std. TPR (%)	Avg. FPR (%)	Std. FPR (%)
SURF	80.53	21.96	62.22	41.46	0.24	0.63
SIFT	76.43	24.22	53.93	46.48	0.15	0.24
HIERARCH-SIFT	74.04	21.25	48.29	42.62	0.2	0.31
Proposed Method	**64.73**	**17.36**	**30.16**	**34.90**	**0.69**	**1.25**
ZERNIKE	63.72	16.04	28.60	29.57	0.23	0.11
ZERNIKE2	60.97	22.44	27.45	41.88	5.50	7.45
KPCA	61.46	16.53	24.06	30.90	0.22	1.35
DCT	61.10	15.89	23.21	29.49	0.10	0.40
DWT	61.09	15.86	23.34	29.41	0.23	1.28
FMT	53.72	10.06	8.42	17.19	0.06	0.34
CIRCLE	53.44	8.63	7.96	213.96	0.15	0.88
LIN	52.33	8.15	5.61	12.74	0.03	0.13
HU	50.40	5.33	2.09	4.26	0.36	1.22
BLUR	49.86	5.07	0.99	2.98	0.36	1.29
SVD	49.85	5.05	0.78	2.79	0.15	0.53
PCA	49.50	4.81	0.02	0.23	0.09	0.34

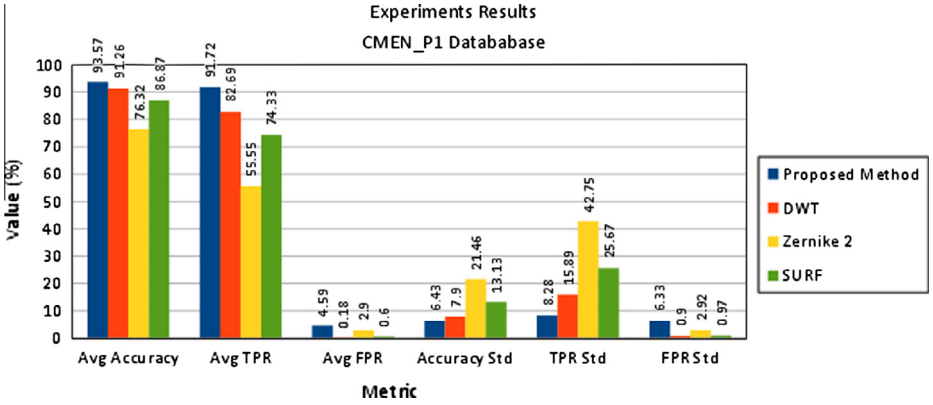


Fig. 9. Comparative results between state-of-the-art methods and the proposed approach for $CMEN_{P1}$ dataset.

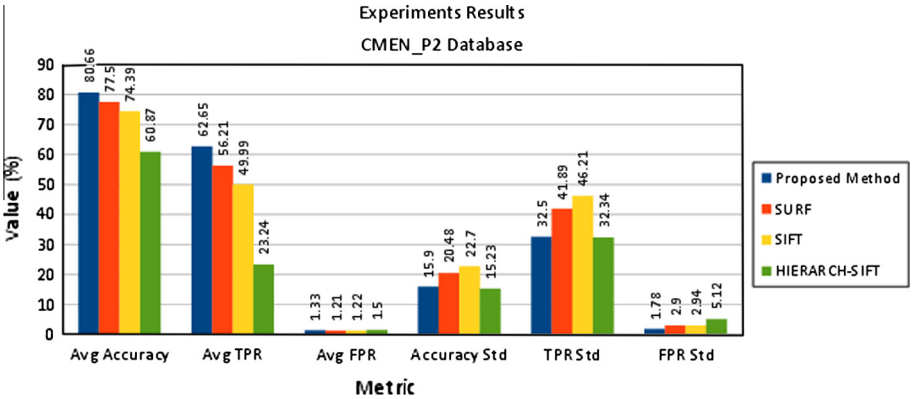


Fig. 10. Comparative results between state-of-the-art methods and the proposed approach for $CMEN_{P2}$ dataset.

4.4. Round #1: Quantitative experiments performed on CMH dataset

This round of experiments evaluated the detection accuracy of the proposed method by a quantitative analysis performed using CMH dataset. The proposed method is compared against 15 state-of-the-art methods previously described in Section 2. Most of these methods have been chosen based on a previous study conducted by Christlein et al. [6]. Other source codes were kindly provided by other authors.[5] To facilitate the reading, for each state-of-the-art method tested in this round of experiments, we associated a label to it as Table 1 shows.[6]

This round of experiments also investigated how sensitive the approaches are to different operations using, separately, each part of the CMH dataset. For each detection map, we compared it to the respective reference map at the **pixel level** and calculated the TPR, FPR, and ACC metrics. Tables 2–7

[5] We gratefully thank Lamberto Ballan for the source code of HIERARCH-SIFT and Seungjin Ryu and Mathias Kirchner for the source code of ZERNIKE2.

[6] To evaluate SIFT performance on copy-move detection, Christlein et al. [6] have used the method proposed by Shivakumar and Baboo [32], replacing SURF by SIFT descriptor. We adopted the same methodology here.

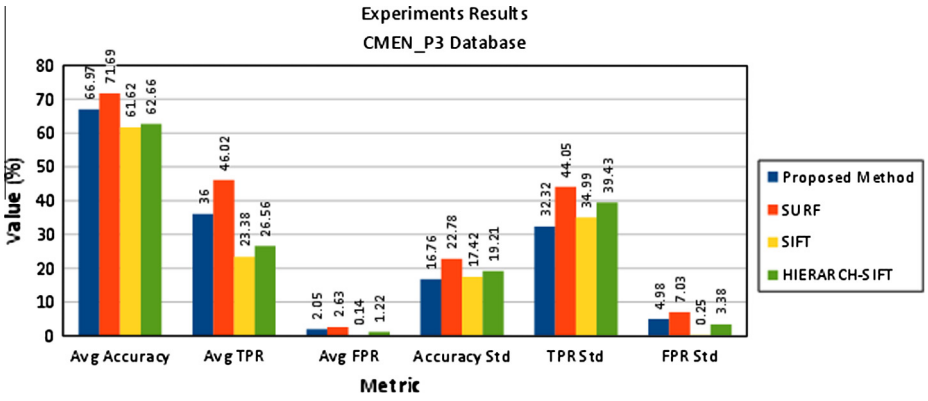


Fig. 11. Comparative results between state-of-the-art methods the and the proposed approach for $CMEN_{P3}$ dataset.

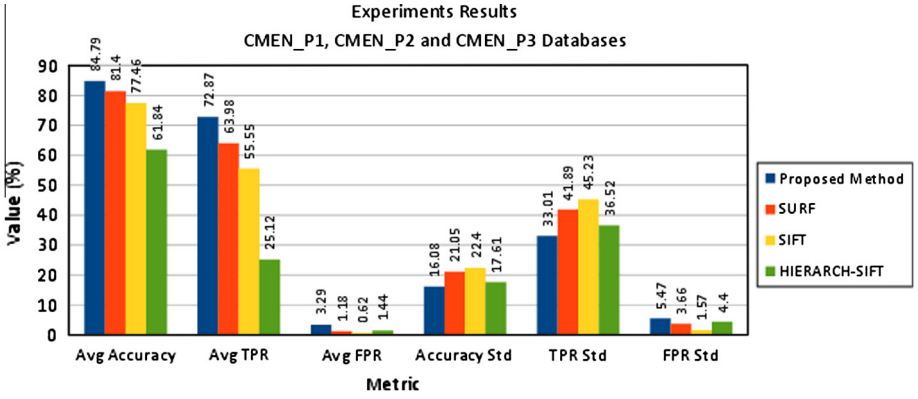


Fig. 12. Comparative results between best state-of-the-art methods and the proposed approach for full CMEN dataset ($CMEN_{P1} \cup CMEN_{P2} \cup CMEN_{P3}$).

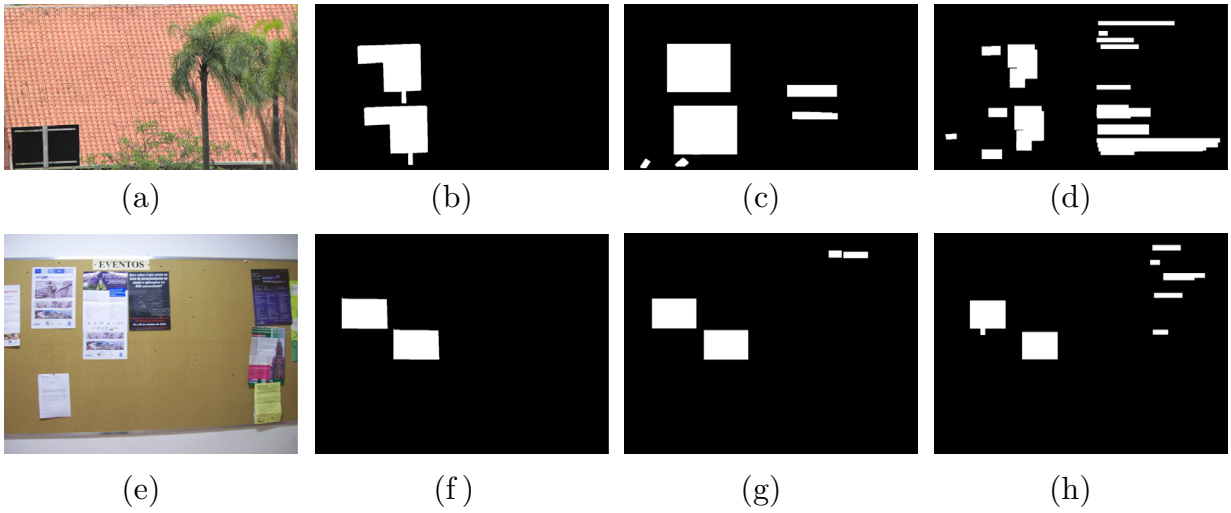


Fig. 13. Qualitative results in simple cloning scenarios. (a) and (e) are images containing cloned segments; (b) and (f) are their groundtruth maps; (c) and (g) are detection maps for uncompressed images and (d) and (h) are detection maps using JPEG compression factors of 90 and 70, respectively.

present results obtained for each individual subset of the CMH dataset.[7]

As showed in Tables 2–6, our approach has achieved the highest ACC, for each one of evaluated CMH subsets, when compared against 15 state-of-the-art methods. This indicates that our method is good to detect copy-move forgeries containing different operations (e.g., rotation and resize) even when these operations are applied together over the same image. Our method also achieved lower standard deviation in ACC when compared to methods with good accuracy as SURF and SIFT. This reduction in variation is important as it is directly related to the robustness of the methods when facing very challenging images. Another relevant result is regarding the TPR metric, for which the minimum

[7] Tables are sorted by average accuracy (Avg. ACC) from the best to the worst method.

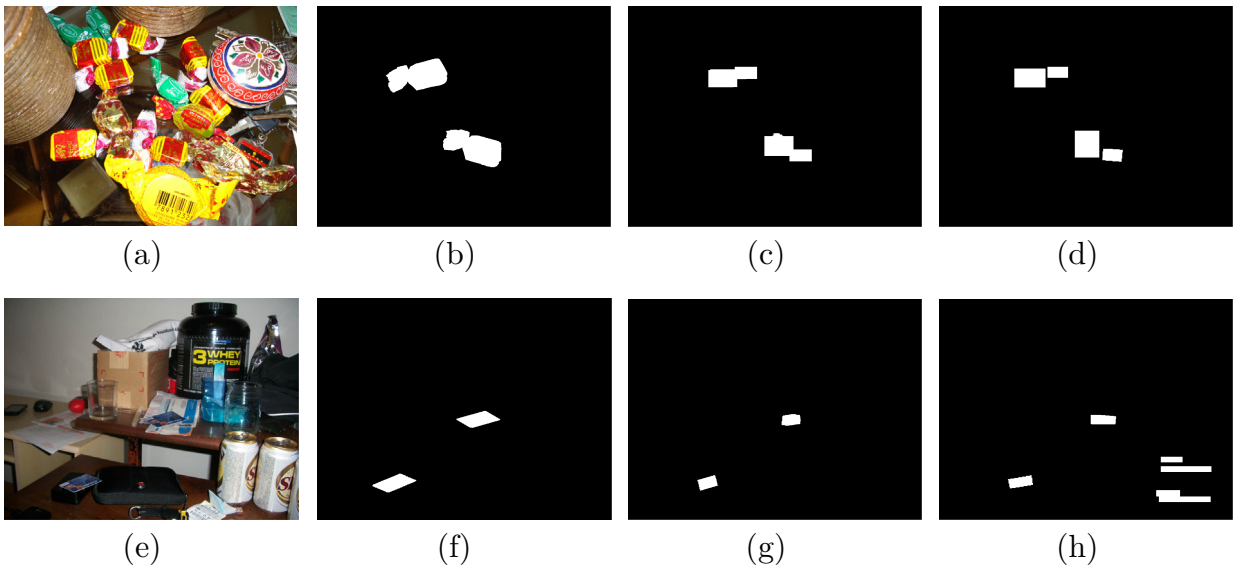


Fig. 14. Qualitative results in rotation scenarios. (a) and (e) are images containing rotated cloned segments (−30 e 7°, respectively); (b) and (f) are their groundtruth maps; (c) and (g) are detection maps for uncompressed images and (d) and (h) are detection maps using JPEG compression factors of 80 and 70, respectively.

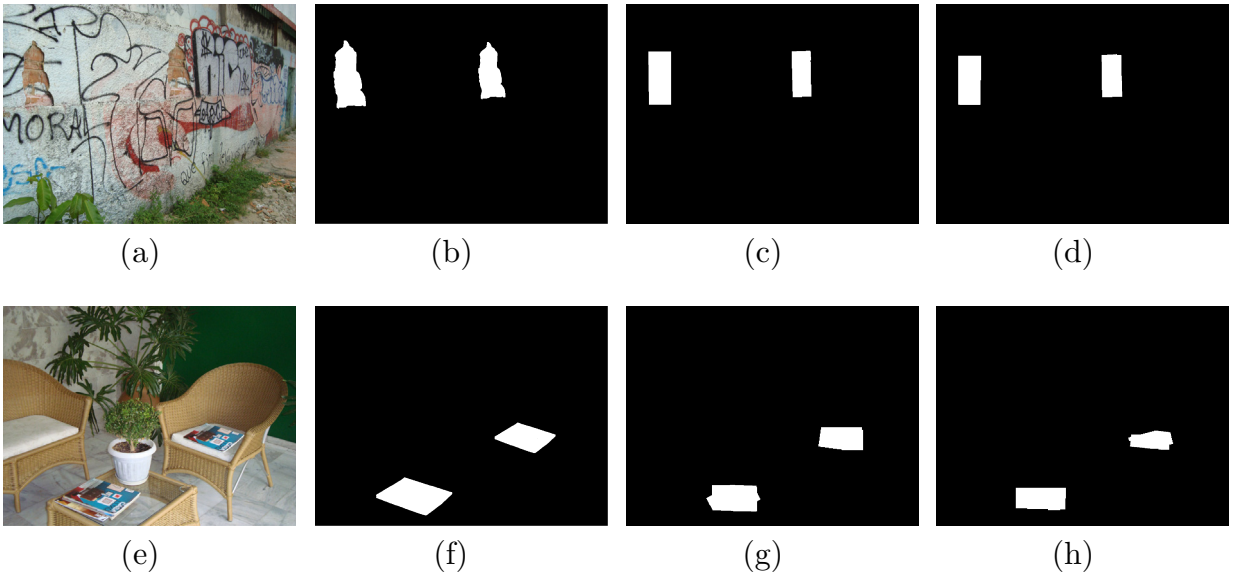


Fig. 15. Qualitative results in resizing scenarios. (a) and (e) are images containing resized cloned segments (upsampling of 19% and downsampling of 20%); (b) and (f) are their groundtruth maps; (c) and (g) are detection maps for uncompressed images and (d) and (h) are detection maps using JPEG compression factors of 90 and 70, respectively.

and maximum values found were 30.16% and 94.08%. Such fact means that even in the worst detection scenario, nearly 30% of the cloned regions area were effectively detected.

Despite the method having slightly higher FPR rate when compared against some state-of-the-art methods, it never exceeds 1.7%, pointing out that it has a good precision for detecting copy-move regions regardless the employed set of operations for creating the forgeries.

Finally, when considering the effects of JPEG compression, Table 7, some interest-point-based techniques such as SURF, SIFT, and HIERARCH-SIFT perform better than the proposed technique. However, the proposed method still achieved the fourth best result, being more effective than 14 state-of-the-art methods.

4.5. Round #2: Quantitative experiments performed on CMEN datasets

In this round of experiments, we evaluate the robustness of the proposed method when evaluated in different datasets. For this, we considered the CMEN datasets previously described. The best three state-of-the-art methods on each operation set (e.g., best three for rotation operation, etc.) from Round #1 experiment have been selected for comparison. For this dataset in particular, we had to resize the images for the Zernike2 approach given that this method was extremely high computationally demanding. In some cases, one image was taking more than one day to calculate. The dimensions chosen were 512×512 because these are the same dimensions used in the image samples given by the authors. Figs. 9–12 depict, respectively, the performance of the methods

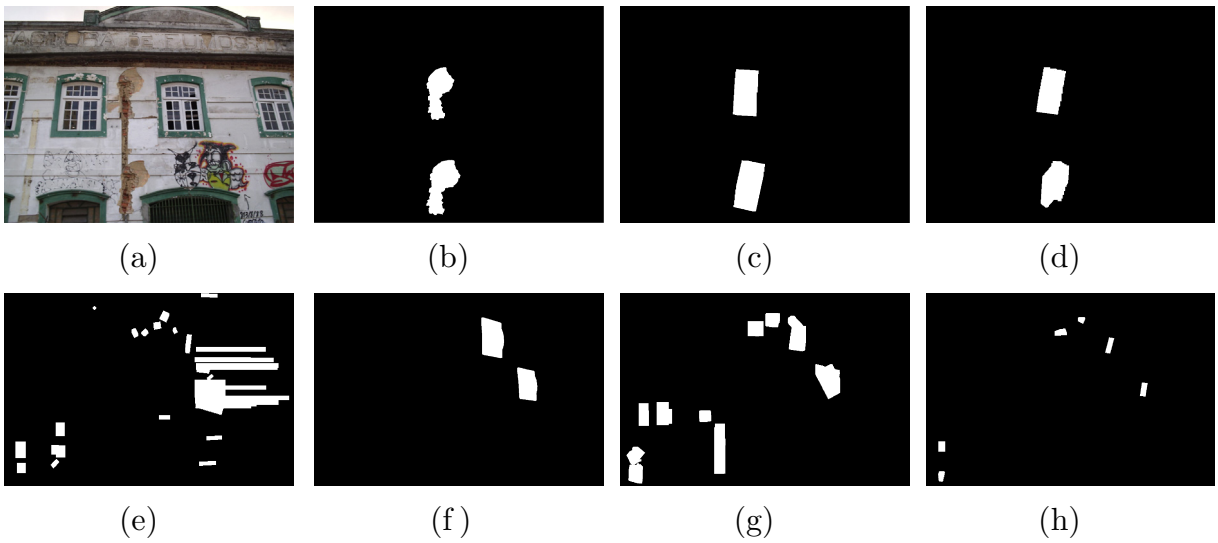


Fig. 16. Qualitative results in composite scenarios. (a) and (e) are images containing resized and rotated cloned patches (−10° and upsampling of 8% and 3° and downsampling of 16% respectively); (b) and (f) are their ground truth maps; (c) and (g) are detection maps for uncompressed images and (d) and (h) are detection maps using JPEG compression factors of 90 and 80, respectively.

when applied on $CMEN_{P1}$, $CMEN_{P2}$, $CMEN_{P3}$, and ($CMEN_{P1} \cup CMEN_{P2} \cup CMEN_{P3}$) according to the metrics used in experiments evaluation.

In accordance to the results found in CMH sets, the proposed method showed superior results in three of four testing scenarios. Just on $CMEN_{P3}$, which comprises scaling operations, SURF presented better performance but at the cost of a larger standard deviation when compared to the proposed method, which is more reliable (smaller variations across different images). Finally, although not shown here, we also performed experiments considering CMEN images when JPEG-compressed. In those cases, our method has slightly worse results than SIFT and SURF and ties in third place with Zernike. However, compared to other methods, it presents the lowest standard deviation showing high reliability for different evaluation conditions.

4.6. Round #3: Qualitative experiments performed on CMH dataset

In this round of experiments, we have qualitatively tested the proposed method with images containing some challenging operations: (i) simple clonings (copy and move, only), (ii) rotations, (iii) resizings, and (iv) JPEG compression. Composite operations of any combination of the above operations were also considered. Figs. 13–16 depict some qualitative results from the proposed approach.

As in other works, there is a drawback in the proposed method caused by the lack of SURF keypoints in some cloned regions. These regions usually have very small size or some homogeneous characteristics, so that the SURF algorithm finds only a small number of keypoints (sometimes no interest point is found at all). In these areas our method might fail.

4.7. Running times

We now turn our attention for an analysis of the efficiency of all evaluated methods. We show the running time of all techniques used in the experiments. For that, we randomly chose 10 images from CMEN Dataset, which are bigger than the ones in CMH dataset, and calculate the mean execution time for analyzing these images for each method. The experiments were performed on a computer with two Intel(R) Xeon(R) E5620 @2.40 GHz CPU with 16 cores and 48 GB RAM. Table 8 shows the results.

Table 8
Mean running time of the proposed technique and each analyzed state-of-the-art method.

Time range	Method	Time (s)
time < 99	SIFT	6.83
	Proposed Method	**18.81**
	LIN	35.76
	HU	36.19
	CIRCLE	39.45
	SURF	42.56
	FMT	55.33
	Zernike	56.40
	SVD	62.52
	DWT	96.60
99 < time < 999	BLUR	113.19
	HIERARCH-SIFT	142.13
	PCA	180.11
	DCT	296.74
	KPCA	880.00
time > 999	ZERNIKE2	1418.68

Clearly, the running times of copy-move detection approaches based on interest points SIFT and the proposed method were lower than the others also based on interest points (HIERARCH-SIFT and SURF) and the ones based on block matching (LIN, HU, CIRCLE, FMT, ZERNIKE, SVD, DWT, BLUR, ZERNIKE2, PCA, DCT and KPCA), explainable by the long time of matching a number of blocks that are close to the number of pixels. The fastest technique among the block matching ones was LIN, because of the radix-sort linear running time employed therein. The slowest technique based on interest points was the HIERARCH-SIFT. The reasons for that are the resulting binary image construction running time and also because it is the only technique not implemented in a compiled language. As for the proposed technique, note that in addition to the excellent results presented regarding the quality of cloning detection it is also a very fast, ranked as the second fastest.

5. Conclusions and future work

In this paper, we performed a deep analysis of the copy-move forgery detection problem by first presenting its main challenges,

as well as several ideas (in the literature) that increased our understanding about the characteristics of the problem. The main aspect we have found out to be incipient in other methods has been regarding to the combination of post-processing operations, although there was still much to be improved when tackling such operations alone.

Our idea introduces new and employs some well-established concepts to address that goal. We explore the suspect image in detail by first changing it to a more suitable color space (HSV), which gives a well-defined color and intensity information of the image pixels. Detecting, matching and clustering interest points (robust to rotation and resizing) in candidate cloning regions are paramount steps, as they spot out and set-up the regions to be compared. In the multi-scale analysis step, we explore these regions and check whether they are cloned or not in several scales. Each region in each scale is examined by using a circular window and a descriptor that characterizes the information inside it. By employing the Euclidean norm to compare the descriptions and decide whether, or not, they represent similar pixel blocks, we obtain an additional robustness to rotation operations. This idea, combined with a voting step, provides a high confidence to the method at effectively separating duplicated regions from false-positives.

The proposed copy-move forgery detection approach works under rotation, resizing, and these operations combined. When using JPEG compression, the proposed method performs well but is not the top-1 when compared to the literature (it is actually the tied-third). When considering all combined operations, the proposed method is highly effective and reliable being the top-ranked in most of the tested datasets with small variations across different evaluation setups compared to 15 other available methods. The slight decreased performance for stronger JPEG-compressed images probably comes from the fact that strong compressions affect the way we compare regions in multiple scales (due to the compression artifacts). That is the price paid for being more robust to scaling, rotation and other operations combined. Finally, the proposed method is also very efficient. Among all techniques evaluated in this paper, it ranks as the second fastest.

Detecting representative interest points in an image, although at the heart of our method, is also responsible for its failure in certain setups. The detector might not find a sufficient amount of keypoints in a small or homogeneous region, which will possibly cause it to be discarded or blocked from further evaluations (if no keypoint is found at all). A more robust strategy to overcome this limitation would include the integration or fusion of other image descriptors to the method, such as SIFT, and a complementary investigation of the regions without keypoints. Other future work comprise an extension of the proposed approach to cope with even more complex operations combined (e.g., blurring, different noise addition, etc.). New strategies to describe block pixels and other ways to detect representative keypoints are some potential improvements as well.

Certainly there is no silver bullet to solve all the problems and we are aware that the proposed method is just a step toward solving the cloning detection problem. However, from interest point detection and clustering, to a multi-scale analysis and a voting process, we believe our method presents several contributions that will provide the literature with new ways to cope with this challenging problem. Finally, with the low false-positive rates found and high reliability regardless the testing scenario (low standard deviations across different testing conditions), we also think our idea may be very useful in a real-world forensics scenario, providing an investigator with a set of suspect cloning regions to be visually checked in the image, as one of the prior steps in determining its authenticity.

Acknowledgments

Supported by Microsoft Research, the São Paulo Research Foundation (Grant #2010/05647-4), the National Council for Scientific and Technological Development (Grants #304352/2012-8 and #477662/2013-7), CAPES (Grants #0214-13-2), CNPq (140916/2012-1), CAPES DeepEyes Project, and IF Sudeste MG.

Appendix A. CMH dataset setup

This appendix presents a listing of the rotation and resizing parameters used to generate the manipulated images in the CMH dataset. It is relevant to inform that the listing does not reflect the amount of images in the CMH dataset, since there are images which have been manipulated with the same rotation and resizing parameters (e.g., two distinct images have a 1° rotation in the cloned area), and this data is shown only once in the respective list.

- Rotation-only by θ degrees
  - $\theta \in$ {−55°, −30°, −16°, 13°, −10°, −9°, 8°, −6°, 1°, 3°, 6°, 7°, 8°, 14°, 28°, 33°, 42°, 75°, 90°, 150°, 180°}
- Resizing-only (downscale or upscale) by scale factor s
  - Downscale $s \in$ {20%, 12%, 10%, 8%, 7%, 5%, 2%}
  - Upscale $s \in$ {3%, 5%, 6%, 8%, 9%, 11%, 13%, 15%, 16%, 18%, 19%, 20%, 23%, 25%, 54%}
- Rotation by θ degrees + Resizing by scale factor s
  - Rotation θ + Downscale s $(\theta, s) \in$ {(−31°, 9%), (−30°, 13%), (−18°, 10%), (−16°, 7%), (−10°, 10%), (−3°, 9%), (3°, 16%), (3°, 5%), (4°, 3%), (13°, 13%), (178°, 5%)}
  - Rotation θ + Upscale s $(\theta, s) \in$ {(−90°, 1%), (−90°, 8%), (−55°, 11%), (−36°, 12%), (−15°, 20%), (−12°, 15%), (−10°, 8%), (−9°, 12%), (−9°, 8%), (−8°, 10%), (−5°, 16%), (−3°, 26%), (−2°, 40%), (2°, 6%), (2°, 7%), (2°, 8%), (4°, 11%), (4°, 31%)(5°, 13%), (5°, 15%), (7°, 3%), (18°, 12%)}

References

[1] E. Silva, A. Rocha, Anlise forense de documentos digitais: alm da viso humana, Saúde, Ética Justiça 16 (1) (2011) 9–17.
[2] H. Farid, Exposing digital forgeries in scientific images, in: ACM Multimedia and Security Workshop (MM&Sec), Geneva, Switzerland, 2006, pp. 29–36.
[3] A. Rocha, W. Scheirer, T.E. Boult, S. Goldenstein, The unseen challenge data sets, in: Intl. CVPR Worktorial on Vision of the Unseen (WVU), 2008, pp. 1–8.
[4] The Joint Photograph Expert Group, JPEG File Interchange Format, July 2010. <http://www.jpeg.org/public/jfif.pdf>.
[5] S. Bayram, H.T. Sencar, N. Memon, An efficient and robust method for detecting copy-move forgery, in: Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP), 2009, pp. 1053–1056.
[6] V. Christlein, C. Riess, J. Jordan, C. Riess, E. Angelopoulou, An evaluation of popular copy-move forgery detection approaches, Trans. Inform. Forensics Secur. (T.IFS) 7 (6) (2012) 1841–1854.
[7] J. Fridrich, D. Soukal, J. Lukas, Detection of copy-move forgery in digital images, in: Digital Forensic Research Workshop (DFRWS), Cleveland, USA, 2003, pp. 134–137.
[8] A.C. Popescu, H. Farid, Exposing Digital Forgeries by Detecting Duplicated Image Regions, Tech. Rep. TR 2004-515, Dept. of Computer Science – Dartmouth College, Hanover, USA, 2004.
[9] W. Luo, J. Huang, G. Qiu, Robust detection of region-duplication forgery in digital image, in: Intl. Conference on Pattern Recognition (ICPR), 2006, pp. 746–749.
[10] B. Mahdian, S. Saic, Detection of copy-move forgery using a method based on blur moment invariants, Forensic Sci. Int. (2006) 180–189.
[11] J. Zhang, Z. Feng, Y. Su, A new approach for detecting copy-move forgeries in digital images, in: Intl. Conference on Communication Systems (ICCS), 2010, pp. 362–366.
[12] G. Li, Q. Wu, D. Tu, S. Sun, A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD, in: Intl. Conference on Multimedia and Expo. (ICME), 2007, pp. 1750–1753.
[13] X.B. Kang, S.M. Wei, Identifying tampered regions using singular value decomposition in digital image forensics, in: International Conference on Computer Science and Software Engineering, 2008, vol. 3, 2008, pp. 926–930. http://dx.doi.org/10.1109/CSSE.2008.876.

[14] S.-J. Ryu, M.-J. Lee, H.-K. Lee, Detection of copy-rotate-move forgery using zernike moments, in: Intl. Workshop in Information Hiding (IHW), 2010, pp. 51–65.
[15] S.-J. Ryu, M. Kirchner, M.-J. Lee, H.-K. Lee, Rotation invariant localization of duplicated image regions based on zernike moments, Trans. Inform. Forensics Secur. (T.IFS) 8 (8) (2013) 1355–1370.
[16] S. Bravo-Solorio, A.K. Nandi, Exposing duplicated regions affected by reflection, rotation and scaling, in: Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011, pp. 1880–1883.
[17] M. Bashar, K. Noda, N. Ohnishi, K. Mori, Exploring duplicated regions in natural images, Trans. Image Process. (T.IP) http://dx.doi.org/10.1109/TIP.2010.2046599.
[18] J. Wang, G. Liu, H. Li, Y. Dai, Z. Wang, Detection of image region duplication forgery using model with circle block, in: Intl. Conference on Multimedia Information Networking and Security (MINES), 2009, pp. 25–29.
[19] H.-J. Lin, C.-W. Wang, Y.-T. Kao, Fast copy-move forgery detection, WSEAS Trans. Signal Process. (WSEAS-TSP) (2009) 188–197.
[20] E. Ardizzone, G. Mazzola, Detection of duplicated regions in tampered digital images by bit-plane analysis, in: Intl. Conference on Image Analysis and Processing (ICIAP), 2009, pp. 893–901.
[21] J.-W. Wang, G.-J. Liu, Z. Zhang, Y.-W. Dai, Z.-Q. Wang, Fast and robust forensics for image region-duplication forgery, Acta Automat. Sinica 35 (2010) 1488–1495, http://dx.doi.org/10.3724/SP.J.1004.2009.01488.
[22] C. Barnes, E. Shechtman, A. Finkelstein, D.B. Goldman, The generalized patchmatch correspondence algorithm, in: European Conference on Computer Vision (ECCV), 2010, pp. 29–43.
[23] C. Barnes, E. Shechtman, A. Finkelstein, D.B. Goldman, Patchmatch: a randomized correspondence algorithm for structural image editing, ACM Trans. Graph. (ToG) (2009) 24:1–24:11.
[24] D. Cozzolino, G. Poggi, L. Verdoliva, Copy-move forgery detection based on patchmatch, in: Intl. Conference on Image Processing (ICIP), 2014.
[25] 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), Computer Society, 2008, pp. 272–276.
[26] D.G. Lowe, Object recognition from local scale-invariant features, in: Intl. Conference on Computer Vision (ICCV), 1999, pp. 1150.
[27] X. Pan, S. Lyu, Detecting image region duplication using sift features, in: Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP), 2010, pp. 1706–1709.
[28] X. Pan, S. Lyu, Region duplication detection using image feature matching, Trans. Inform. Forensics Secur. (T.IFS) 5 (4) (2010) 857–867.
[29] I. Amerini, L. Ballan, R. Caldelli, A.D. Bimbo, G. Serra, A sift-based forensic method for copy-move attack detection and transformation recovery, Trans. Inform. Forensics Secur. (T.IFS) 6 (3) (2011) 1099–1110.
[30] B. Xu, J. Wang, G. Liu, H. Li, Y. Dai, Image copy-move forgery detection based on surf, in: Intl. Conference on Multimedia Information Networking and Security (MINES), 2010, pp. 889–892.
[31] H. Bay, T. Tuytelaars, L. Van Gool, Surf: Speeded up robust features, in: European Conference on Computer Vision (ECCV), 2006, pp. 404–417.
[32] B.L. Shivakumar, S. Baboo, Detection of region duplication forgery in digital images using SURF, Int. J. Comput. Sci. Issues 8 (2011) 199–205, http://dx.doi.org/10.3724/SP.J.1004.2009.01488.
[33] Intel, The OpenCV Reference Manual 2.4.3, November 2012. <http://docs.opencv.org/opencv2refman.pdf>.
[34] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, Trans. Pattern Anal. Mach. Intell. (T.PAMI) 27 (10) (2005) 1615–1630.
[35] Y.-M. Wong, Invariant Local Feature for Image Matching, December 2006. <http://www.cse.cuhk.edu.hk/lyu/student/mphil/wyman/term3paper.pdf>.
[36] R. Szeliski, Computer Vision: Algorithms and Applications, first ed., Springer, 2011.
[37] G. Bradski, A. Kaehler, Learning OpenCV, first ed., O'Reilly Media, 2008.