EURASIP Journal on Image and Video Processing
a SpringerOpen Journal

RESEARCH Open Access

Counter-forensics of SIFT-based copy-move detection by means of keypoint classification

Irene Amerini[1], Mauro Barni[2], Roberto Caldelli[1] and Andrea Costanzo[2*]

Abstract

Copy-move forgeries are very common image manipulations that are often carried out with malicious intents. Among the techniques devised by the 'Image Forensic' community, those relying on scale invariant feature transform (SIFT) features are the most effective ones. In this paper, we approach the copy-move scenario from the perspective of an attacker whose goal is to remove such features. The attacks conceived so far against SIFT-based forensic techniques implicitly assume that all SIFT keypoints have similar properties. On the contrary, we base our attacking strategy on the observation that it is possible to classify them in different typologies. Also, one may devise attacks tailored to each specific SIFT class, thus improving the performance in terms of removal rate and visual quality. To validate our ideas, we propose to use a SIFT classification scheme based on the gray scale histogram of the neighborhood of SIFT keypoints. Once the classification is performed, we then attack the different classes by means of class-specific methods. Our experiments lead to three interesting results: (1) there is a significant advantage in using SIFT classification, (2) the classification-based attack is robust against different SIFT implementations, and (3) we are able to impair a state-of-the-art SIFT-based copy-move detector in realistic cases.

1 Introduction

Counter-forensics, the study of methods to mislead forensic techniques by concealing traces of manipulations, is becoming a hot research topic [1]. As a matter of fact, this discipline rapidly became a benchmark for the security of image forensic techniques, whose correct behaviors may be intentionally obstructed by an adversary (or attacker) interested on covering traces of malicious tampering. Nowadays, for example, it is possible to conceal traces of contrast enhancement, compression, or resampling [2-4]. In [1], two categorizations of counter-forensic schemes (or attacks) are proposed: according to when, in the image acquisition chain, an attack takes place (*integrated* or *post-processing*, i.e., during or after the acquisition) and according to whether the countered forensic algorithm is known or unknown to the attacker (*targeted* or *universal* attack, respectively). Regardless of its category, an attack should respect some constraints while attempting to mislead a certain forensic technique, such as preserving the visual quality of the forged image and the integrity of the semantic message conveyed by the content.

Copy-move forgery, whereby a portion of the image is copied and pasted once or more times elsewhere into the same image, is one of the most common ways of manipulating the semantic content of a picture. Usually, the original and the forged portions share similar statistical and semantic properties. As a consequence, it is possible to exploit such similarities to detect the presence of the manipulation. In the past years, literature has proposed several copy-move detectors [5], which often are conventionally summarized into two categories: methods based on blocks and methods based on keypoints. The former methods first divide the image into overlapping blocks and then extract some features which are ordered and used to match similar blocks, according to certain criteria of similarity [6,7]. The latter methods extract highly descriptive robust points of an image, to each of which a univocal features vector is assigned. Such vectors are then used to match similar points across images (or regions within them). Among such methods, the most recent and effective ones [8,9] are those based on scale invariant feature transform (SIFT) [10]. The capability of SIFT to discover correspondences between similar visual contents, in fact,

*Correspondence: andreacos82@gmail.com
[2]Department of Information Engineering and Mathematical Sciences, University of Siena, Siena 53100, Italy
Full list of author information is available at the end of the article

© 2013 Amerini et al.; licensee Springer. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

allows the forensic analysis to detect even very accurate and realistic copy-move forgeries.

The research community has recently started to approach copy-move detection from the perspective of the attacker, whose goal is to hide the features causing similar blocks or keypoints to match. Chrislein et al. [11] studied the robustness of several detectors against common image processing and observed that block-based methods are not robust against geometric manipulations (e.g., resampling, cropping), lossy compression, and noise addition. Nguyen et al. [12] successfully impaired three well-known block-based detectors by combining some of these manipulations into a simple yet effective counter-forensic scheme (see [12]). The same results, however, cannot be replicated with SIFT-based detectors inheriting the intrinsic robustness against geometric transformations from Lowe's algorithm. To devise more sophisticated schemes, it is necessary to understand the security of SIFT algorithm. The first study in this direction is the one by Hsu et al. [13], in which, initially, the impact of simple attacks is analyzed, and then a method to strengthen SIFT keypoints is proposed. Following this work, Do et al. [14-16] focused on a SIFT-based content-based image retrieval (CBIR) [17] scenario and devised a number of interesting attacks.

The aim of the previous works is to modify the SIFT feature descriptor of keypoints, but they do not consider the complete removal of the keypoints. To the best of our knowledge, the only work in this sense is [18], where an attack based on local warping techniques derived from image watermarking was proposed. All the studies carried out so far have demonstrated that devising methods to attack SIFT features is not a trivial task. SIFT features are robust not only against several nonmalicious processing, but also against tampering attempts. Most attacks, in fact, pay a high cost in terms of visual quality degradation. Moreover, the attempt to remove SIFT keypoints can alter the content in such a way that new keypoints are created, thus complicating even more the problem.

The methods for countering SIFT-based forensic analysis proposed so far have been applied indifferently to all the keypoints of an image. In this paper, we present a new counter-forensic strategy (targeted and post-processing, according to the terminology of [1]) that permits to improve the performance of the existing approaches. We demonstrate that it is possible to discriminate between SIFT keypoints and to devise attacks that are tailored to the characteristics of the keypoints.

Specifically, we have chosen a classification criterion based on first-order statistics, which is the pixel histogram of the gray scale neighborhoods centered in the SIFT keypoints. On top of such classification, we have built an iterative attacking algorithm. At each step, our algorithm classifies the keypoints and then attempts to delete them with class-tailored attacks, ideally until their complete removal.

We compared the proposed method against other attacks [13-16] and demonstrated the benefits brought by SIFT classification in terms of keypoint removal and visual quality. We also demonstrated that the proposed method is also capable of impairing SIFT detectors different than the one used during the attack. We then applied our method to a realistic scenario of copy-move forgery detection with the goal to disable the detector described in [9]. We have successfully completed this task by eliminating only the keypoints having matches across the copy-moved regions that were used to detect the manipulation. Finally, we have explored the possible interactions between the proposed method and the existing block-based copy-move countermeasures.

The paper is organized as follows: section 2 briefly reviews the SIFT algorithm and the copy-move detectors based on it. Section 3 describes the rationale underlying the SIFT classification and the algorithm which has been used to perform it. Section 4 introduces the framework, putting in practice the principles previously introduced. Section 5 experimentally validates the framework by means of three important results of the proposed method: (1) the advantage of SIFT classification with respect to blind attacks, (2) the robustness against different implementations of the SIFT algorithm, and (3) the effectiveness in a copy-move detection scenario. Section 6 concludes the paper.

2 SIFT-based copy-move forgery detection

In this section, we briefly review the SIFT technique and describe the copy-move detectors based on the technique.

2.1 Scale invariant feature transform

SIFT features have become extremely popular in pattern recognition applications due to their robustness with respect to partial occlusion, clutter, and geometric transformations [10]. The idea behind this kind of visual local features is to model a complex object or a scene by a collection of salient points.

In a nutshell, SIFT features of an image are detected at different scales using a scale-space representation implemented as an image pyramid. The pyramid levels are obtained by Gaussian smoothing and sub-sampling of the image resolution, while interest points are selected as local extrema (min/max) in the scale space.

The detection of extrema usually produces numerous candidate keypoints. However, not all the candidates possess the robustness and the stability required to become a keypoint, and thus, they need to be discriminated. To this end, the SIFT algorithm performs two checks against two different thresholds, whose commonly accepted values were experimentally set by Lowe in [10]. The first check verifies whether the contrast value of the keypoint's neighborhood is sufficiently large. The second check verifies whether a keypoint is distant enough from an image edge (edges are considered unstable). If either of the checks fails, the candidate keypoint is rejected. It is worth noting that such thresholds represent the principal vulnerability of the SIFT algorithm, as we will see in section 4. Attacks against SIFT, in fact, locally alter the image in such a way that either a real keypoint falls below one of the thresholds (false negative) or a fake keypoint raises above one of the thresholds (false positive).

The keypoints that passed both the previous tests guarantee invariance to scaling and affine transformations. At this point, the algorithm assigns to each of them a canonical orientation in order to also guarantee rotation invariance. This task is performed by means of a histogram of gradient orientations computed in the neighborhood of the keypoint (defined by a specific window). Finally, a unique fingerprint, called descriptor, is computed in order to identify univocally a keypoint. Therefore, a SIFT keypoint is completely described by the following information: $\mathbf{x}_i = \{x, y, \sigma, o, \mathbf{f}\}$, where (x, y) are the coordinates in the image plane, σ is the scale of the keypoint, o is the canonical orientation, and $\mathbf{f}$ is the final SIFT descriptor.

2.2 Copy-move detection

In pattern recognition, the SIFT operator usually is applied to two images: a target and a test image. In the case of copy-move forgery detection, the SIFT operator is applied to one image only. In fact, in a copy-move forgery, the copied part is within the same image (see images of Figure 1). For this reason, the keypoints extracted in that region will be quite similar to the original ones; therefore, a matching between SIFT features can be used to discover which part was copied and which geometric transformation was applied.

In the past years, different techniques have been proposed by the scientific community to address the problem of copy-move forgery detection in digital images. Most of the methods divide the image into overlapping blocks and then extract some peculiar features that can reveal whether some of the blocks have been duplicated or not [5]. Depending on the amount and on the characteristics of the paired blocks, a decision about forgery is then made. Unfortunately, such methods are hardly robust against rotation and scaling operations, which are very common in copy-move forgeries. The SIFT features allow to overcome such limitations, thanks to their intrinsic robustness against geometric transformations. Among the most recent SIFT-based copy-move detection techniques, we are interested in the one proposed by Amerini et al. [9], which is able to detect and estimate the geometric transformation applied in a copy-move forgery attack also dealing with the case of multiple copy-move forgeries.

In a nutshell, the technique in [9] works as follows (see Figure 1). Given an image I, the method extracts the keypoints $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and their SIFT descriptors $D = \{f_1, \ldots, f_n\}$. The best candidate match for each keypoint $\mathbf{x}_i$ is found by identifying its nearest neighbor among the other $n-1$ keypoints, i.e., the keypoint with the minimum Euclidean distance descriptors.

For the sake of clarity, given a keypoint, a similarity vector $S = \{d_1, d_2, \ldots, d_{n-1}\}$ is defined with the sorted Euclidean distances with respect to the other descriptors. The keypoint is matched only if $^{d_1}/_{d_2} < T$ (fixed empirically to 0.6). By iterating on each keypoint in X, a set of matched points is obtained.

Although this set already provides a draft idea of the presence of cloned areas, a clustering procedure is run in order to improve accuracy. To understand whether an area has been cloned or not, an *agglomerative hierarchical clustering* is performed on spatial locations, i.e., (x, y) coordinates, of the matched points. Such method creates a hierarchy of clusters which can be represented by means of a tree structure. Briefly, the clustering algorithm works as follows: (1) each keypoint is assigned to a cluster, (2) the reciprocal spatial distances among clusters are computed, (3) the closest pair of clusters is found, and (4) the obtained pair is merged into a single cluster. Consequently, if two (or more) clusters are detected with at least 4 pairs of matched points linking a cluster to another, then the corresponding regions are considered cloned. When an image has been classified as non-authentic, the method can also determine which geometric transformation was applied between the original area and its copy-moved version by employing an affine homography.

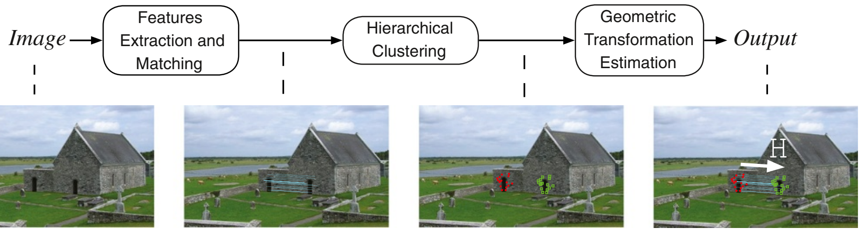
Image
Features Extraction and Matching
Hierarchical Clustering
Geometric Transformation Estimation
Output
H

Figure 1 Overview of the method proposed in [9]. SIFT matched pairs and clustering.

3 Classification of SIFT keypoints

In this section, we first describe the classification method, then we introduce the classes that we have defined, and finally, we provide some visual examples of each class.

3.1 The rationale behind the classification

In principle, we would imagine that the classification relies on the visual content surrounding the keypoints. Although this task could be performed in more than one way (e.g., textures, edges, shapes), we chose to analyze the gray scale histogram of a relatively small region surrounding the keypoint. More specifically, among all the characteristics of an image histogram, we chose the number of modes since they provide valuable information about the local image content. The idea is that, in general, the effectiveness of an attack may be strictly related to the properties of the keypoint we attempt to remove. As an example, suppose that the neighborhood of a keypoint contains a straight vertical edge; a local warping attack, such as the one in [18], would probably succeed in deleting it. Unfortunately, after the attack, the edge would not be straight anymore and the bending effect would be clearly visible. Perhaps a Gaussian smoothing attack may delete the keypoint as well, arguably with a significantly lower impact on quality.

For the rest of the paper, we will make two assumptions: we will work on gray scale images and will consider only the SIFT keypoints originated by the first scale of the image ($s = 0$). The reasons behind the latter assumption are the following: (1) the keypoints of the first octave are the most difficult to remove, and (2) for the sake of clarity, we work with a significant yet not excessive amount of keypoints.

3.2 SIFT keypoint classification algorithm

In order to classify the keypoints, we have adapted the histogram analysis method proposed by Chang et al. in [19], which was originally designed for image segmentation based on histogram thresholding. The original algorithm relies on the assumption that the histogram of large natural gray scale images can be modeled as a mixture of Gaussians f as follows:

$$f(k) = \sum_{i=1}^{n+1} \frac{P_i}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2}\left(\frac{k-m_i}{\sigma_i}\right)^2} \tag{1}$$

where $k = 1 \ldots 256$ are the samples of the mixture, $n+1$ is the number of histogram segments, and (P_i, m_i, σ_i^2) are respectively the weight, the mean, and the variance of the i-th Gaussian. The classifier is designed to estimate the model parameters in order to minimize $|f - H|$, where H corresponds to the original histogram. In a nutshell, it proceeds as follows. First, H is smoothed to reduce the number of unstable local extrema. Then, the local minima are determined and the segments between consecutive pairs of minima are initialized. For each segment i, the calculation of the parameters (P_i, m_i, σ_i^2) consists of two steps: the estimation of a unique optimal window w^* with minimum skewness near the center of the segment and the estimation of the Gaussian parameters in w^*, which are then refined by means of a maximum likelihood criterion. Finally, the thresholds used to segment the histogram are computed by relying on the refined Gaussian parameters.

Here, we are interested more in the number of modes rather than in the thresholds for segmenting H. We cannot rely directly on the number of Gaussians composing the mixture since Chang et al.'s algorithm tends to over-segment the histograms, thus creating rather flat segments with very small weight. Therefore, the original technique required some adjustments in order to fit our application. Given a keypoint, the classification was modified as follows: (1) we initialize the number of modes equal to the $n+1$ contributes of the mixture (where n has been computed by means of Chang's method), (2) we estimate all the Gaussian parameters and determine the weight of the largest contribution, namely $P_{\max}$, and (3) we suppress all the contributions $1 \leq i \leq n+1$ such that $P_i \leq 0.2 \cdot P_{\max}$, thus obtaining the number of histogram modes M. It goes without saying that if $M = 1$, the histogram is considered unimodal, if $M = 2$ bimodal and if $M > 2$ multimodal.

The choice of the size N of the neighborhood used for the classification is closely related to Chang et al.'s algorithm. N should be large enough in such a way that the hypothesis of Gaussianity holds. At the same time, it should not be so large to include the statistics of content that is too far from the support of the keypoint. According to our experiments, a good trade-off is obtained by letting $N = 32$.

3.3 Classes of keypoints

By relying on the method described previously, we have classified the 32×32 neighborhoods of several thousands of keypoints extracted from natural images with different visual content (landscapes, people, buildings). Our observations confirmed that the histograms tend to cluster into well-defined groups. We defined three of them: unimodal, bimodal, and multimodal (number of modes > 2). Interestingly, these classes correspond to very different visual contents: uniform flat regions with low variance tend to have a unimodal histogram, edges and geometric shapes correspond to bimodal histograms, and regions with high variance (which resemble some sort of noise) usually have a multimodal histogram. Figure 2 provides an example of a keypoint belonging to each of the three classes. The first row shows the visual content of the neighborhood; the second row shows its gray scale histogram. For this example, we have used an image representing a landscape.

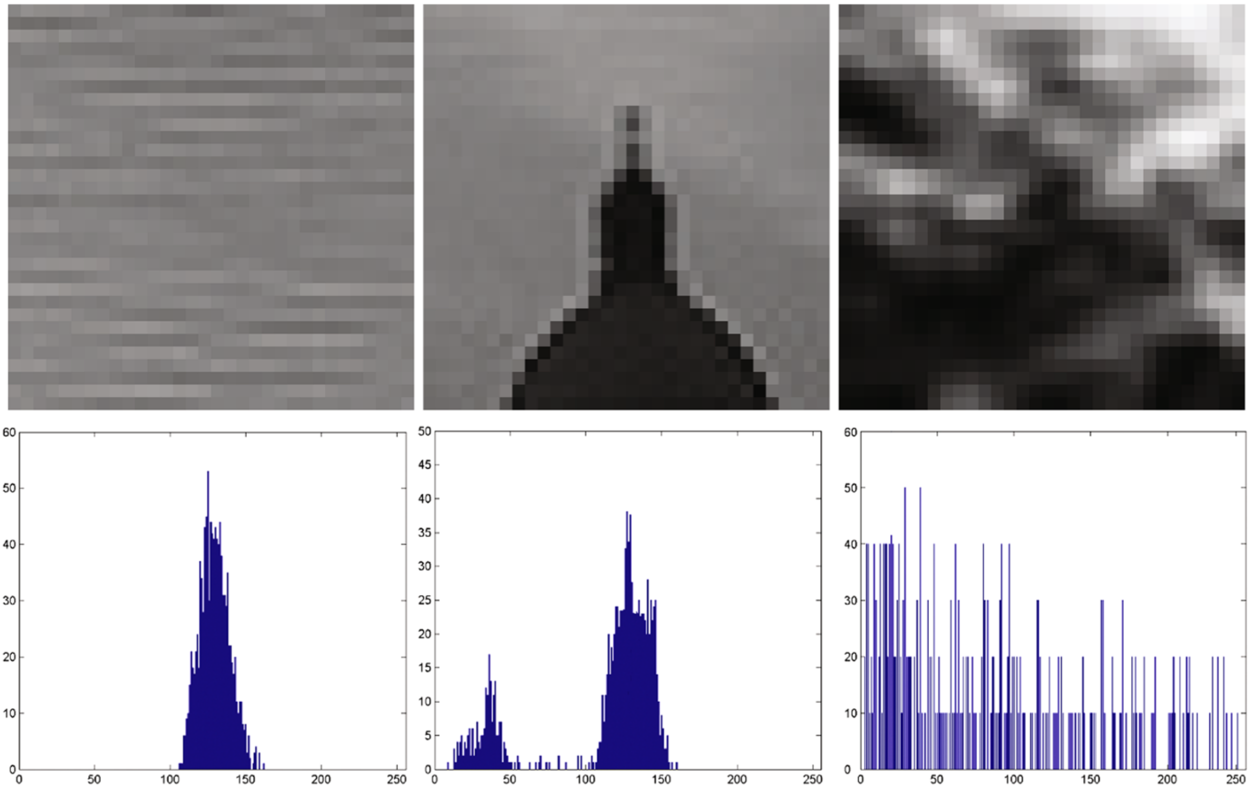

Figure 2 Example of visual contents (first row) and histograms (second row) for the three classes of keypoints. From left to right: unimodal, bimodal, and multimodal.

The unimodal, bimodal, and multimodal contents represent respectively a water surface, the top of a building, and some foliage.

4 The attack

In this section, we first describe the various attacks and then explain how they have been combined to remove SIFT keypoints and to invalidate the copy-move detection algorithm. Even if the classifier works on a fairly large neighborhood of the keypoint, the attacks are carried out on an 8 × 8 region (still centered on the keypoint), in order to reduce the visual impact of each attack. In the following, we will refer to this region with the term 'patch'.

4.1 The proposed framework

Before discussing in depth the attacks, it may be useful to briefly introduce our framework (see Figure 3). It relies on an iterative procedure. At the beginning (first iteration), an original gray scale image G (or a region within it) is fed to the system, which starts by detecting the SIFT keypoints, and then, for each keypoint, the neighborhood of size 32 × 32 centered on the keypoint is extracted and classified accordingly to its gray scale histogram. Depending on the class of each keypoint, the corresponding 8 × 8 patch is manipulated by means of a class-tailored attack. Finally, the manipulated patches are inserted back into the image in their original positions. The procedure moves to the next iteration and halts only when particular requirements are met (e.g., percent of deleted keypoints, maximum iterations, minimum allowed visual quality).

4.2 Single attacks

The first attack is the *smoothing attack*. A light Gaussian smoothing flattens the pixel values of an image in such a way that its potential keypoints at the level of difference of Gaussians (DoG) are reduced. On the other hand, an excessive smoothing has a very noticeable impact on the visual quality. The intensity of the attack can be controlled with the parameters (h, σ), i.e., the size and the standard deviation of the Gaussian kernel. In our experiments, we have found out that $h = 3$ and $\sigma = 0.7$ represent a good compromise between the amount of deleted keypoints and the overall visual quality following the attack. This attack has also been used in [15].

The second attack is the *collage attack*, which is a variant of the method first used in [13]. It consists on the substitution of the original patch with another patch of the same size. The new patch should not contain a keypoint and needs to be as similar as possible to the original one according to some similarity criteria. To implement

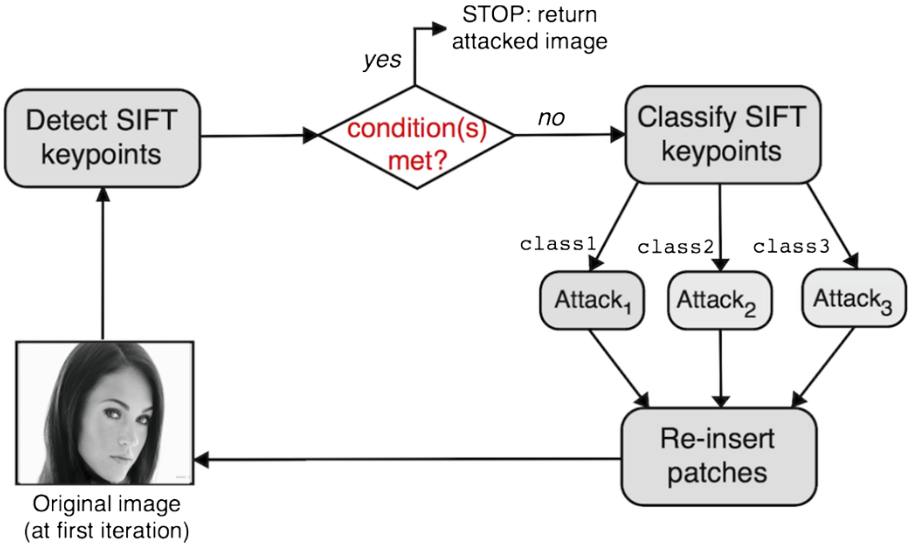

Figure 3 Proposed attack scheme. SIFT keypoints are detected and classified; the neighborhood of each keypoint is attacked with a specific class-tailored attack and then inserted back into the image. The procedure continues until the specified conditions are met.

the attack, we created a database of about 120,000 patches not containing keypoints, extracted from a data set of 80 images characterized by very heterogeneous visual contents. We chose to measure the similarity by means of the histogram intersection distance, which has been widely used in the past in image retrieval applications [20]. Let H_{orig} and H_{db} be, respectively, the histograms of the original patch and of a patch stored in the database; the intersection distance d_{int} is evaluated as follows:

$$d_{\text{int}}(H_{\text{orig}}, H_{\text{db}}) = \frac{\sum_{j=1}^{L} \min\left(H_{\text{orig}}(j), H_{\text{db}}(j)\right)}{\sum_{j=1}^{L} H_{\text{db}}(j)}, \qquad (2)$$

where j indicates the j-th bin of the histogram and $L = 256$ indicates the number of bins. Now, let patch$_{\text{orig}}$ and patch$_{\text{min}}$ be respectively the original patch and the most similar counterpart stored in the database (i.e., the patch whose histogram is at minimum d_{int}); to avoid visible artifacts along the borders, we do not reinsert patch$_{\text{min}}$ directly into the original image. Instead, we reinsert the following linear combination:

$$\text{patch}_{\text{new}} = W \cdot \text{patch}_{\text{orig}} + (1 - W) \cdot \text{patch}_{\text{min}} \qquad (3)$$

where W is an empirical 8 × 8 weighting matrix, whose elements $w_{i,j} \in [0, 1]$ are set to 1 along the patch borders and progressively decrease to 0 near the center, as shown in Figure 4.

The third attack is the *removal with minimum distortion (RMD) attack* proposed by Do et al. in [15]. The idea behind this technique is to calculate a small patch ϵ that, added to the neighborhood of a keypoint, allows its removal. The coefficients of ϵ are chosen in such a way that the contrast around the keypoint (at DoG level) is reduced, thus invalidating the check performed by SIFT algorithm on all potential keypoints. Moreover, it is requested that the coefficients locally introduce the minimum visual distortion.

In a nutshell, the *RMD* attack works as follows. Let $\mathbf{x} = (x, y, \sigma)$ be a keypoint and let $D(\mathbf{x})$ be the DoG in $\mathbf{x}$; the patch ϵ is obtained by resolving the following optimization problem:

$$\epsilon = \arg\min_{\epsilon : D'(\mathbf{x}) = D(\mathbf{x}) + \delta} \left(||\epsilon||^2 \right), \qquad (4)$$

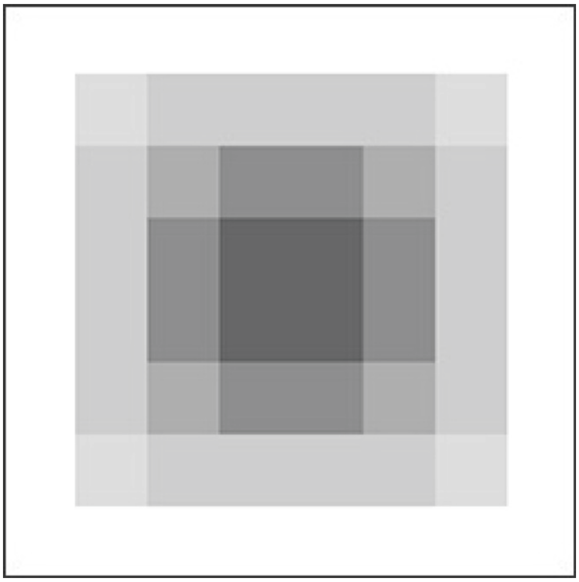

Figure 4 Weighting window 8 × 8 of Equation 3. The coefficients decrease from 1 (white) along the borders to 0 (black) near the center.

where δ is a parameter that allows to control the intensity of the attack. Let C be the contrast threshold set by the SIFT algorithm (usually by default $C = 0.03$); the attack reduces $|D(\mathbf{x})|$ by $|\delta|$ in such a way that the altered value drops below C.

The size of ϵ depends on the spatial support of the targeted keypoint: the larger the support, the stronger the attack. The final altered DoG region is then $D(x + u, y + v, \sigma)$, with both u and v belonging to the interval $[-6\sqrt{2}h\sigma, 6\sqrt{2}h\sigma]$ and $h = 2^{\frac{1}{3}}$. To be compliant with our system, we introduced two small variants into the original method: first, we limited the size of ϵ to a maximum of 8×8 also for those keypoints with larger spatial support; second, we used the same weighting window of Equation 3 to replace the original neighborhoods.

4.3 Combined attack

Sometimes, an attack may introduce new keypoints in its attempt to delete those already present (see [14]). In such cases, a single iteration of the attack is not enough since, now, one needs to deal also with the newly introduced keypoints. For this reason, we arranged our attacks into an iterative procedure with a pseudocode illustrated in the following (see Algorithm 1). The purpose of the attack is to iteratively remove SIFT keypoints until one of the following requirements is met: the algorithm reaches the maximum number of allowed iterations (maxIter); the desired number of keypoints has been removed from the image (minRemoval, ideally 100%). Practically, at each iteration, we compute the keypoints and classify them. For the first part of the iterations (1 to $K = 10$ by default), we attack all the classes with *smoothing* attack, while in the second part ($K + 1$ to maxIter), we attack the unimodal and multimodal keypoints by means of *collage* attack and the bimodal ones by means of *RMD* attack. From now on, to measure the effectiveness of an attack (or of a single iteration of it), we will use the term *removal rate*, that is, the percent of keypoints that have been removed from an image with respect to their original number before the attack.

The rationale behind the algorithm is the following. The *smoothing* attack can be effective on all the classes, regardless of their content, since it reduces the population of keypoints without a significant loss of quality. The keypoints that survive to this first round of the attack are somehow 'harder' to remove and require more powerful countermeasures (i.e., *collage* and *RMD*), and here, the keypoint classification plays its basic role. The *collage* is not suitable for those patches that contain geometric edges (i.e., the bimodal ones) since the histogram similarity does not take into account the shapes contained in the keypoint neighborhood. Therefore, we manipulate the bimodal keypoints with *RMD*, which does not present this problem. On the contrary, the *collage* attack can be applied both to uniform patches, such as the unimodal, and to noisy patches, such as multimodal, without an excessive visual quality degradation.

Before we move to the next section, it is important to point out that, in the attempt to remove a keypoint, an attack could obtain the opposite result, that is, to alter the image in such a way that a new keypoint is generated. In our implementation, we chose not to keep track of this different category of keypoints but rather to deal with them in the same way as the original ones. This means that the keypoints introduced during an iteration i are classified and attacked again at the following iteration $i + 1$.

Algorithm 1

```
procedure CLASSIFICATION_BASED_ATTACK(originalImage, maxIter, minRemoval)
    $j \leftarrow 1$
    $K \leftarrow 10$
    $removal\_rate \leftarrow 0$
    attackedImage ← originalImage
    while ($j \leq maxIter$ and $removal\_rate < minRemoval$) do
        keypoints = calculateSIFT( attackedImage )
        kp_classes = classifySIFT( keypoints )
        if ($j \leq K$) then
            attackedImage ← smoothingAttack( kp_classes )
        else
            attackedImage ← collageAttack ( unimodal, multimodal )
            attackedImage ← RmdAttack( bimodal )
        end if
        removal_rate ← calculate_removal_rate( originalImage, attackedImage )
        $j \leftarrow j + 1$
    end while
end procedure
```

5 Experimental analysis

The goal of this section is twofold: first, to highlight the benefits introduced by the classification step with respect to a *class-unaware* attack; while performing such task, we also demonstrate the effectiveness of the attack procedure against various SIFT software; second, to address a specific forensic scenario by countering the copy-move detection technique described in [9].

5.1 Image data sets

The experimental analysis has been carried out on two distinct sets of images. To demonstrate the effectiveness of our technique and its robustness against different SIFT implementations, we have used the UCID database [21]. Such data set, which is a well-known benchmark amongst the image retrieval research community[a], consists of 1, 338 uncompressed (TIFF) color images, with contents depicting landscapes, cityscapes, people, and man-made objects. The rather large size of this collection allowed us to make conclusive statements on the performance of the proposed technique. Finally, to demonstrate the capability of our method to impair a SIFT-based copy-move detector, we have used a set of 10 images containing a realistic copy-move forgery.

5.2 Classification-based SIFT keypoints removal

In the following tests, the keypoints have been computed by means of `VLFeat`, the Vedaldi, and Fulkerson's implementation of SIFT [22] (DoG peak and edge thresholds set respectively to 4 and 10). The 1, 338 images composing the UCID database, which we used for our experiment, contain a total of 385, 750 keypoints at the first octave, distributed as shown in the histogram of Figure 5.

We organized the experimental procedure as follows. We set a target removal rate of 100% (i.e., perfect removal) and a maximum number of iterations (i.e., maxIter= 40). We then separately attacked each image by means of four methods: the classification-based attack of section 4.3, the iteration of *RMD* ($\delta = 2$), the iteration of *smoothing* ($\sigma = 0.7, h = 3$), and the iteration of *collage*. We iterated each attack until we reached either 100% removal rate or the 40-th iteration. Once the algorithm halted, we evaluated the removal rates actually achieved on each image. We organized these values in four histograms, whose envelopes are shown in Figure 6 (results for removal rates below 50% are omitted for the sake of clarity).

Two observations can highlight the superior performance of the classification-based method: (1) it grants a minimum removal rate of 80% practically on the whole data set; (2) in general, it provides the highest removal rate. As an example, let us focus our attention on removal rates greater than 90%: the classification-based method achieved such goal on 1, 149 images out of 1, 338 (corresponding to 86% of the data set), followed by *collage*, which achieved the same results on 468 images (35% of the data set). *RMD* and *smoothing* proved to be the less effective attacks, with 147 (11% of data set) and 12 (0.9% of data set), respectively. It is also worth noting that only the classification-based approach was able to reach the perfect removal, although only on a limited number of images (32, corresponding to 2.7% of the data set).

The number of deleted keypoints is not the only important metric for evaluating the performance of the attacks.

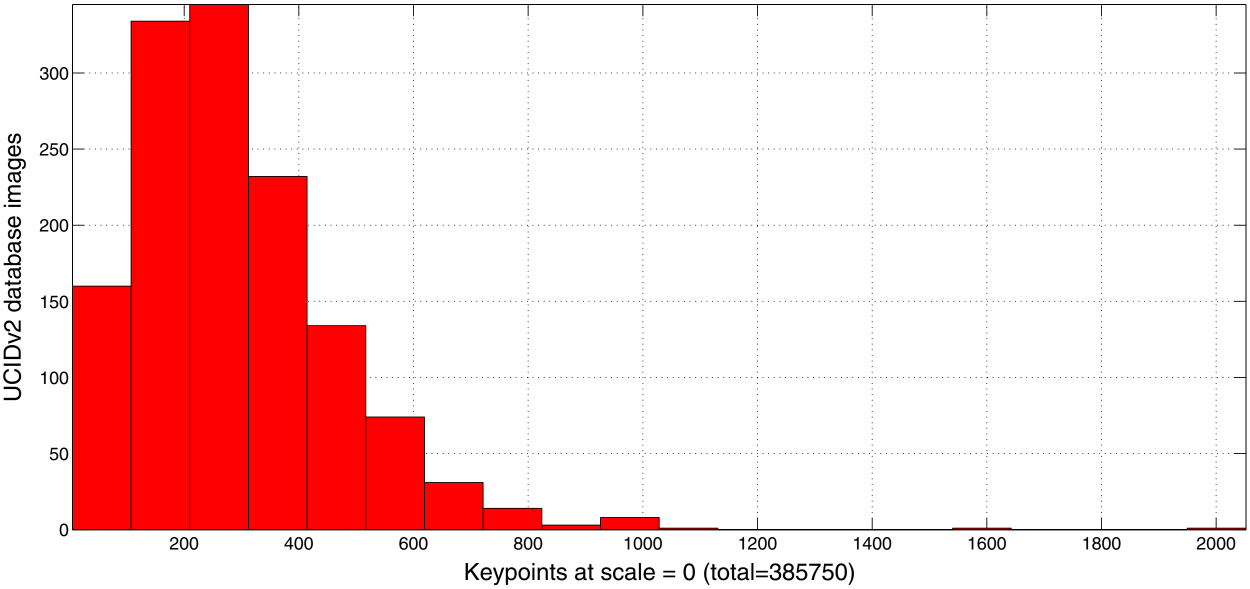

Figure 5 Histogram of the number of keypoints at first octave for images belonging to UCID database.

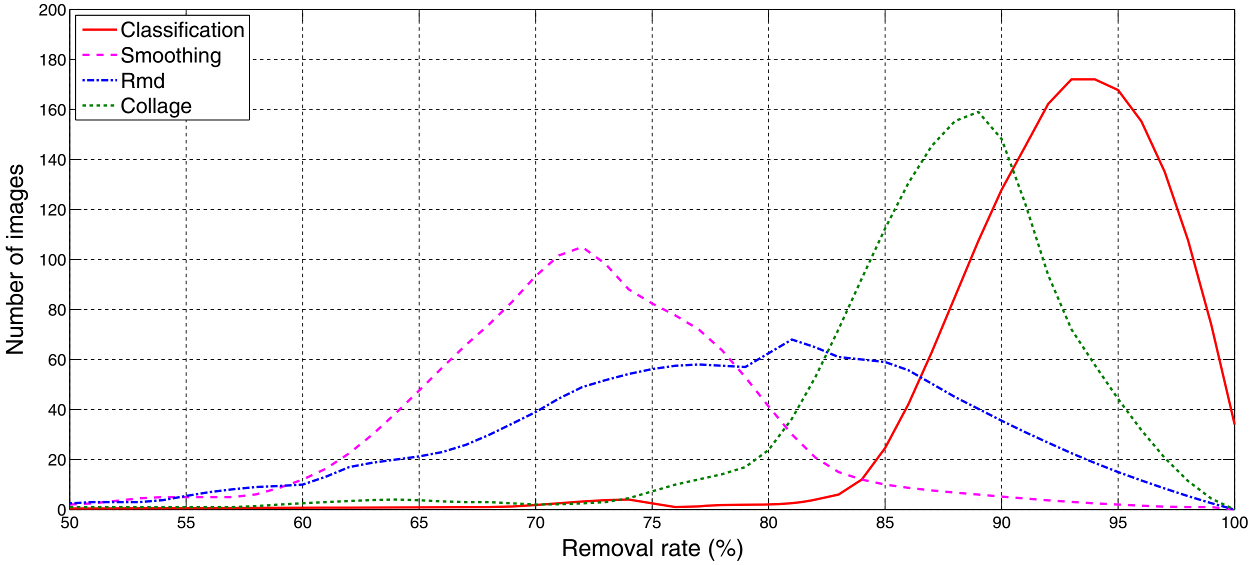

Figure 6 Effectiveness of classification-based attack with respect to the state of the art of *class-unaware* techniques (UCID database).

To be really effective, in fact, an attack also needs to preserve the image quality. Therefore, to evaluate the impact on visual quality, we selected the images belonging to each bin of the histograms that led to Figure 6, and we have averaged the peak signal-to-noise ratio (PSNR) over all the attacked patches. The results for high removal rates are summarized in Table 1, where we omitted rates greater than 90% since averaging on a small number of images did not produce significant results. Clearly, *smoothing* is the attack with the lowest impact on image quality, but such an advantage comes at the price of the lowest removal rates. Amongst the remaining techniques, the classification-based method provides the best quality. One may wonder about the causes behind the poor performance of *collage* and *RMD*: for the former, they may be related to the size or the quality of the database or to the similarity criterion; as for the latter, they are undoubtedly related to the nature of the attack itself. *RMD*, in fact, although very powerful, covers the original patches with very unpleasant 'dots' rather than replacing them with something more similar content-wise. Consequently, this effect quickly deteriorates the local quality, specially for those keypoints with large spatial support.

Figure 7 provides a visual comparison of the three most effective methods. The artifacts introduced by *RMD* (third from left) and *collage* (fourth from left) are more noticeable than those inserted in the picture by the classification-based attack (second from left). Such phenomena are particularly visible between the ears of the dog.

Finally, it is also interesting to evaluate the computational burden of each technique. Within a single iteration of an attack, the main contribution to the time complexity comes from cycling through all the keypoints, while the detection of the SIFT features generally has a negligible impact. In Figure 8, the average execution time for a single image is shown: the higher complexity of the *collage* attack (triangular marker) is the consequence of several comparisons with the database of patches not containing keypoints and becomes evident for removal rates above 75%. The comparisons are still required by the classification-based attack (star marker) but limited in terms of iterations (25 instead of 40) and of classes of keypoints (2 instead of 3). All the tests have been performed on Matlab® on a desktop configuration with 2 GHz dual-core processor, 4 GB RAM, 32-bit Windows OS.

5.3 Robustness against different SIFT implementations

There exist a number of different implementations of the algorithm, often performing differently in terms of results such as the number or the spatial location of keypoints.

Table 1 Average patch PSNR (dB) vs removal rate for the four attacks

	60%	65%	70%	75%	80%	85%	90%
Classification	36.66	36.60	36.59	36.57	36.56	36.55	36.40
Smoothing	41.83	41.79	41.87	41.98	41.85	41.28	40.55
RMD	29.57	29.46	29.31	29.01	28.77	28.40	27.99
Collage	30.34	30.31	30.27	30.24	30.12	29.87	29.71

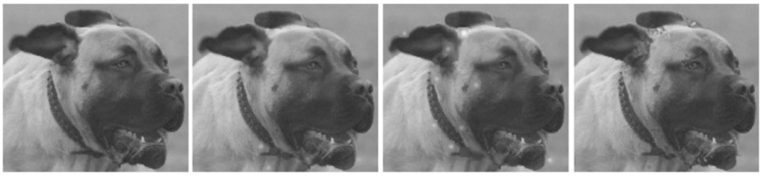

Figure 7 Detail of an attacked region. From left to right: original, *proposed* (98% removed with average patch PSNR of 37.8 dB), *RMD* (94%, 25.6 dB), and *collage* (91%, 35.1 dB). Results refer to the whole image.

Although it is not the aim of this work to assess the fidelity to Lowe's original code of the various implementations, these software should not be ignored when evaluating the robustness of our attack. In fact, one may rightly wonder whether the proposed approach is only capable of impairing the adopted implementation of SIFT by exploiting its weaknesses (and its specific parameters). This should not happen in a realistic counter-forensic scenario, where the attacker typically does not know which SIFT implementation the forensic analyst will be using.

Consequently, to evaluate the robustness of the classification-based attack, we have selected the following four SIFT implementations:

- `VLFeat` [22]. To the best of our knowledge, this is the 'reference library' for the SIFT-based forensic and counter-forensic related works. Other than the present work, [9,14-16] are also based on it. This software is written in C language and can be downloaded from http://www.vlfeat.org.
- `SIFT Legacy` [23] (also known as MATLAB/C and SIFTC++). Although this is basically the predecessor of `VLFeat` and has been superseded by it, such software is still used (see [18]). It can be downloaded from http://www.vlfeat.org/~vedaldi/code/sift.html.
- `Rob Hess SIFT library` [24]. It is written in C and uses the well-established OpenCV computer vision library. It can be downloaded from http://blogs.oregonstate.edu/hess/code/sift/. There also exists a Java porting of this code.
- `Jift` (by Jun Liu). Arguably the less famous implementation, it is written in C++ and uses the VXL computer vision library. It can be downloaded from http://www.cs.man.ac.uk/~liuja/#downloads. There also exists an OpenCV porting of this code.

5.4 Test of robustness

In the next experiments, we attacked again the 1, 338 images of the UCID database but only by means of the classification-based attack (maxIter = 40, minRemoval = 100). Our goal now is to evaluate the behavior of the proposed method in those cases where the attacker and the forensic analyst may be relying on different versions

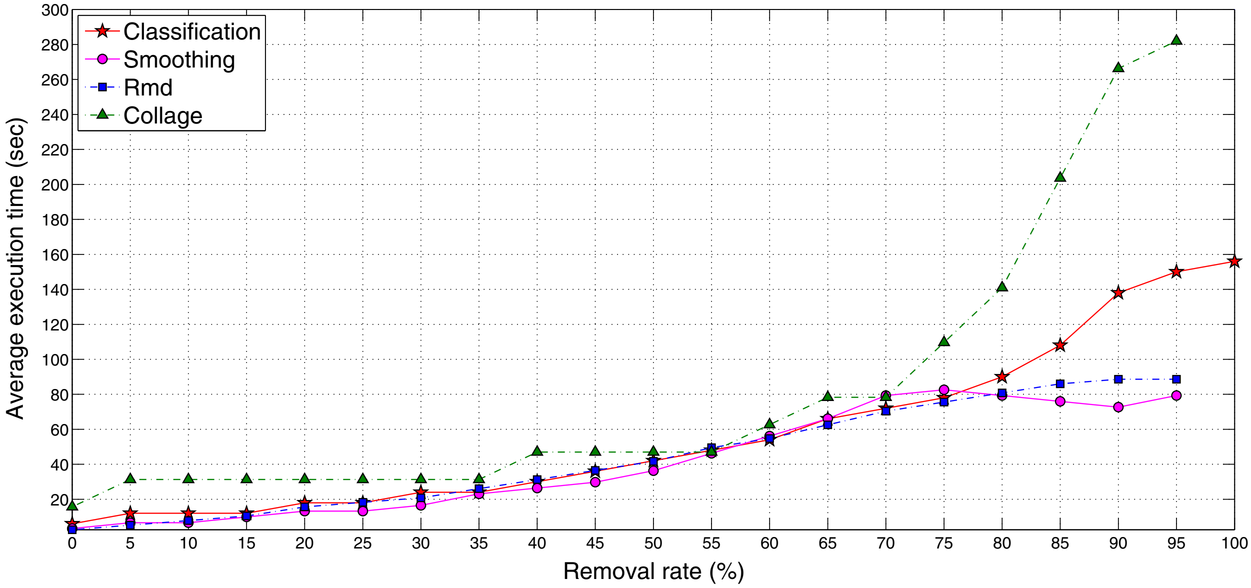

Figure 8 Average processing time for a single image vs removal rate on the UCID database.

Table 2 Main parameters of the employed SIFT implementations

	Octaves			Thresholds		
	Number	Initial	Intervals	Contrast	Edge	Peak
VLFeat	1	0	3	×	10	4
SiftLegacy	1	0	3	0.03	10	4
RobHess	1	0	3	0.04	10	0.8
Jift	1	0	3	0.03	10	0.8

In Vedaldi and Fulkerson's VLFeat the minimum amount of contrast to accept a keypoint is controlled by the peak threshold (hence the × symbol in the table).

of SIFT. In order to obtain fair results, the attack procedure was not tweaked to the characteristics of the various detectors: the only constraint we imposed to the tools at our disposal is to work on the first octave, in compliance with our starting assumptions. We left unchanged all the other parameters (see Table 2) with values that correspond most of the times to those suggested by Lowe in [10].

For each image, we proceeded as follows: (1) we computed the original keypoints with all the SIFT implementations, (2) manipulated the image with the VLFeat classification-based attack, and (3) evaluated the removal rate according to the number of final keypoints detected by each version of SIFT. Similarly to the procedure that led to plots of Figure 6, we organized all the values into histograms, whose envelopes are shown in Figure 9.

As one may expect, the best results were achieved against the VLFeat-based detector (91.8% average removal), followed by SiftLegacy (80%) and Jift (75%). Unfortunately, our method did not seem to be effective against RobHess: averagely, only 9.5% of the detected keypoints were removed. Furthermore, on 216 images out of 1, 338 (about 16% of data set) new keypoints were introduced by the attack's iterations (see negative removal rates of Figure 9). A more accurate analysis revealed that this specific implementation of SIFT often tends to calculate several keypoints in spatial locations different from those detected by the remaining tools. Therefore, the classification-based attack was not actually carried out on the keypoints that RobHess is calculating, whose neighborhoods remained unaltered.

By relying on such new information, we improved the base framework of section 4.1: rather than employing just one SIFT detector, we tried to combine both VLFeat and RobHess detectors during the classification-based attack. As a consequence, at each iteration we classified and attacked the union of the keypoints provided by the two tools. The curves of Figure 10 were obtained by following the same procedure of the previous case.

Not only the attack is now dramatically more effective against the RobHess implementation (76.7% average removal), but the performance remained basically the

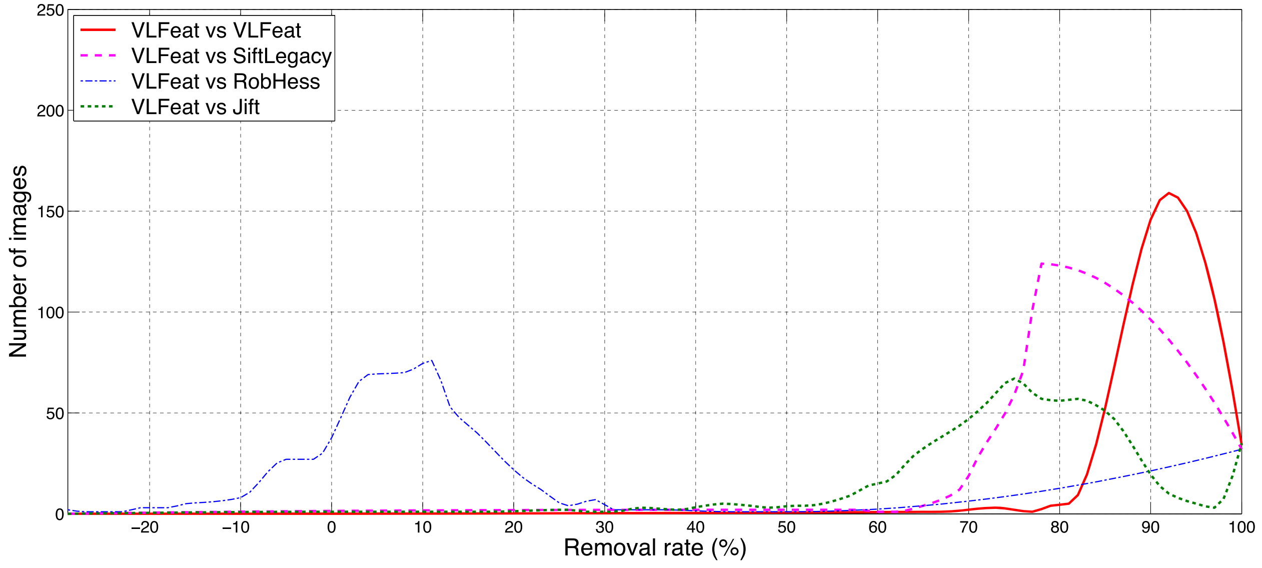

Figure 9 Robustness of the VLFeat-based proposed method. Curves correspond to the envelopes of removal rate histograms, obtained by analyzing the manipulated UCID database by means of four different SIFT versions.

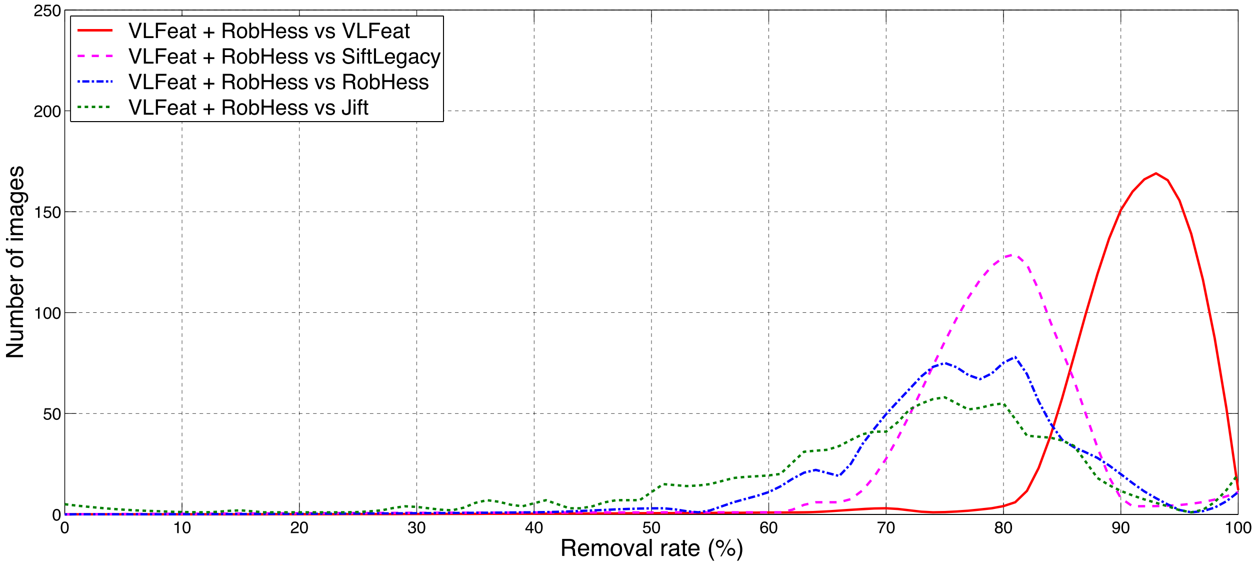

Figure 10 Robustness of the VLFeat+RobHess-based proposed method. Curves correspond to the envelopes of removal rate's histograms, obtained by analyzing the manipulated UCID database by means of 4 different SIFT versions.

same against VLFeat (91.5%), SiftLegacy (79.8%) and Jift (71%). From the latter test, we can conclude that it may be worth investigating the case in which an attacker can select a combination of more than one SIFT detector before removing the keypoints since this would bring the attacker and the analyst on the same level, thus opening new interesting scenarios.

As an example, let us consider the *face2.jpg* test image shown in Figure 11 (left). On the original image, VLFeat and RobHess detect respectively 58 and 24 keypoints. If we attack the image with the classification-based method relying only on VLFeat, we obtain the results of Figure 12 (top), where we omitted iterations 26 to 40 because the number of keypoints remained constant. We can see that, according to VLFeat, there is only 1 keypoint left in the attacked image. However, if we let RobHess analyze the same image, 15 keypoints are detected. Let us now carry out again the attack, this time relying on both VLFeat and RobHess (Figure 12 bottom). Clearly, the number of keypoints is higher now (82), as they are the union of the keypoints detected by the two implementations. However, the attack shows the same trend as before, but this time, it is also effective against RobHess.

It is interesting to point out that this improvement did not come at the cost of visual perceptivity. As Figure 11 can confirm, the quality loss of the combination of VLFeat and RobHess with respect to VLFeat alone, in terms of PSNR of the final image, is barely noticeable: 44.12 vs 45.84 dB for the full image and 36.56 vs 37.75 dB for the average over all the patches.

5.5 Copy-move counter-forensics

In a copy-move scenario, the aim of an attacker is to avoid the detection of matched SIFT keypoints linking the cloned patches. For the sake of simplicity, we can assume to deal with two copy-moved patches without loss of generality. Therefore, our objective is now to hide the traces of the manipulation by altering only the source and destination patches. As a consequence, we will now deal with a significantly lower number of

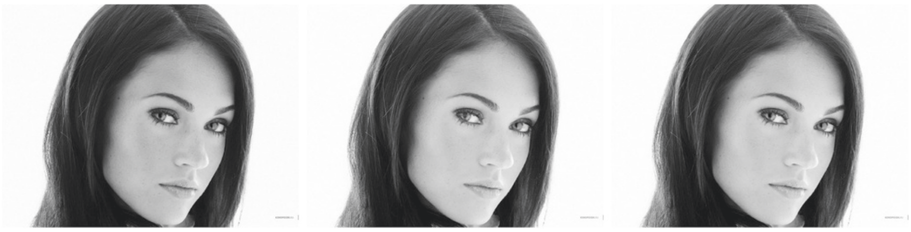

Figure 11 Visual comparison of the impact of the classification-based attack on test image *face2.jpg* (40-th iteration). From left to right: original, VLFeat only, and VLFeat+RobHess.

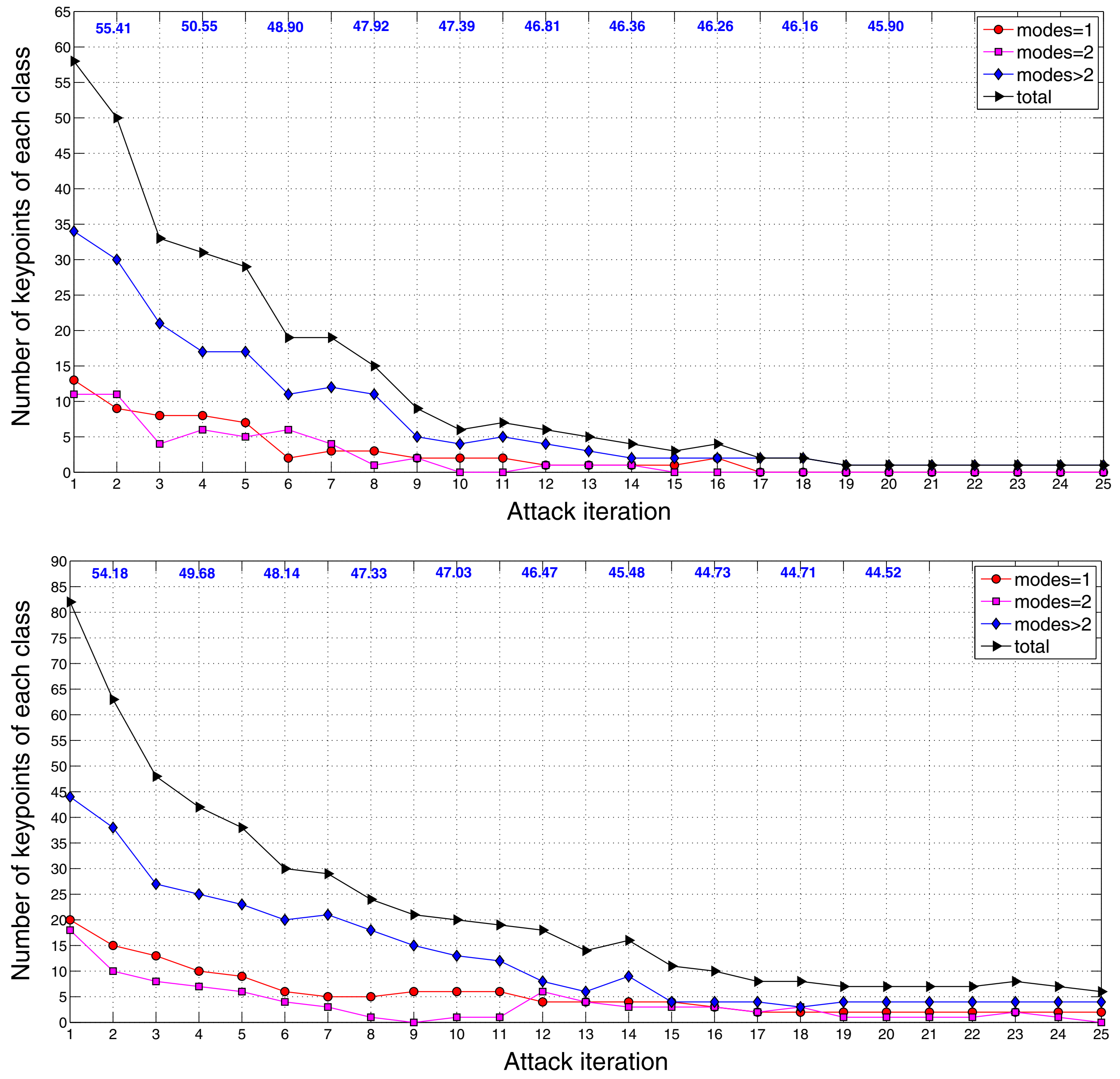

Figure 12 Number and classes of keypoints for each iteration of the attack on test image *face2.jpg*. `VLFeat` only (top) and `VLFeat+RobHess` (down). In the latter case, the total amount of keypoints is the union of the keypoints detected by the two techniques. The values in the upper portion of each plot correspond to the PSNR (full image) of even iterations.

keypoints with respect to the attack carried out on the whole image. As a matter of fact, if the manipulation is wisely distributed over the two patches in such a way that a mismatch in their SIFT descriptions is introduced, one does not even need to delete all the keypoints of the patches.

To this aim, we have slightly modified the procedure of section 4.3 in such a way that, at each iteration, only one keypoint of each match at a time is manipulated. Let N be the number of matches revealed by the copy-move detector; first, we randomly choose $\frac{N}{2}$ matches and try to erase them by attacking the corresponding keypoints in the source patch; we then select the remaining $\frac{N}{2}$ matches and try to erase them by attacking the corresponding keypoints in the destination patch. Although we did not exploit such advantage here, it is interesting to point out that it is not always strictly necessary to completely remove all the matched keypoints. Copy-move detection methods, in fact, usually require at least three matches to detect a forgery.

Table 3 Tests on 10 copy-move manipulated images

	Attack											
Image(matches)	Classification-based			Smoothing			RMD			Collage		
	After (%)	Iter	PSNR	After (%)	Iter	PSNR	After (%)	Iter	PSNR	After (%)	Iter	PSNR
I_1 (37)	0 (*100%*)	14	41.2	12 (67.6%)	40	39.4	5 (86.5%)	40	30.7	3 (91.9%)	40	28.6
I_2 (66)	0 (*100%*)	20	39.9	29 (56.1%)	40	47.7	21 (68.2%)	40	25.6	6 (90.9%)	40	33.7
I_3 (150)	0 (*100%*)	23	48.1	103 (31.3%)	40	57.8	0 (*100%*)	26	41.2	7 (95.3%)	40	43.1
I_4 (41)	0 (*100%*)	15	42.8	11 (73.2%)	40	42.1	9 (78.0%)	40	27.4	2 (95.1%)	40	32.2
I_5 (23)	0 (*100%*)	31	47.0	8 (65.2%)	40	44.5	8 (65.2%)	40	25.3	0 (*100%*)	9	28.0
I_6 (56)	0 (*100%*)	17	45.8	27 (51.8%)	40	41.1	7 (87.5%)	40	26.3	3 (94.7%)	40	33.3
I_7 (55)	0 (*100%*)	28	34.3	10 (81.8%)	40	40.9	10 (81.8%)	40	26.2	1 (98.2%)	40	28.4
I_8 (32)	0 (*100%*)	15	45.0	8 (75.0%)	40	38.7	1 (96.8%)	40	30.1	2 93.8%)	40	29.4
I_9 (52)	0 (*100%*)	19	45.8	21 (59.6%)	40	48.6	10 (80.8%)	40	25.3	2 (96.2%)	40	33.5
I_{10} (53)	0 (*100%*)	21	44.2	16 (69.8%)	40	46.7	22 (58.5%)	40	29.4	3 (94.3%)	40	32.0
Average	*100%*	21	43.4	*63.1%*	40	45.1	*80.3%*	39	28.8	*95.0%*	37	32.2

Iter, iteration. Left matches are listed according to the relative iteration.

Table 3 confirms that the conclusions drawn for the general case of section 5.2 still hold for the copy-move scenario. The classification-based method represents again the best trade-off between the rate of removal (all matches deleted with the lowest number of iterations) and the perceptual quality (average PSNR of 35.2 dB which is second only to the *smoothing* attack).

Figure 13 shows image I_7's copy-moved regions following the four attacks. It can be observed that the detector is fooled by the classification-based attack (see Figure 13 - bottom right image) because the duplication of the skyscraper is not recognized. On the contrary, the detector is still able to reveal the forgery in the other three cases: *smoothing* (10 matches, 40.9 dB), *collage* (1 match, 28.4 dB), and *RMD* (10 matches, 26.2 dB).

5.6 Block-based and SIFT-based copy-move detection

Copy-move forgery detection is not carried out only by means of SIFT-based methods. For this reason, in this

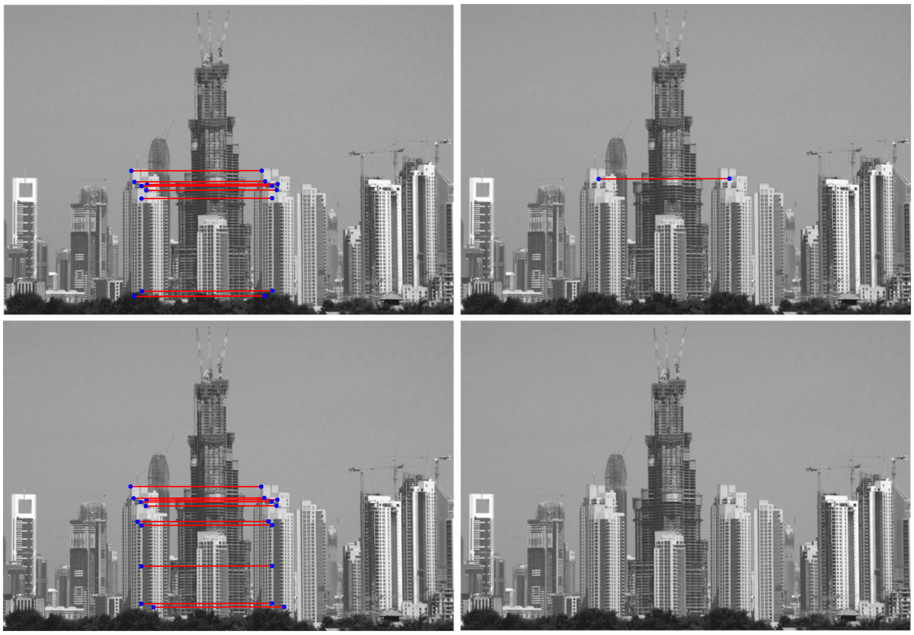

Figure 13 Matches of the copy-moved regions of image I_7 after the four attacks. Top left: *smoothing*; top right: *collage*; bottom left: *RMD*; bottom right: *classification-based*.

section we propose a brief qualitative analysis of the performance of the classification-based attack in the presence of a block-based approach. More specifically, we have employed the following two detectors: the block-based one devised by Fridrich et al. in [6], which relies on the similarity of low frequency DCT coefficients, and the SIFT-based one of [9]. We selected the forged image of Figure 14a, where a person has been removed from the picture by duplicating a portion of the sandy region. Starting from it, we hid the copy-move forgery by means of the following counter-forensic techniques: the classification-based attack and the geometric attack proposed in [12]. According to the authors, the attack of [12] proved to be effective against a number of block-based detectors (including [6]). It consists of a crop of 3 pixels (column-wise and row-wise), followed by two JPEG compressions (qualities 70 and 60), and a final resampling (bicubic interpolation) back to the size of the original image.

We ran the two detectors on the manipulated images, and the results we obtained are show in Figure 14: the first image of each row represents a manipulation, while the second and third images represent the detection performance of the block-based and SIFT-based tools, respectively. Both techniques succeeded in revealing the presence of tampering before the attacks were carried out. Following the geometric attack (Figure 14d), only the block-based detector was successfully impaired. Vice versa, the classification-based attack (Figure 14g) hid the manipulation only to the SIFT-based detector. Unlike the single application of either of the two countermeasures, their cascade (Figure 14l) was effective against both detectors, regardless of the order of the attacks.

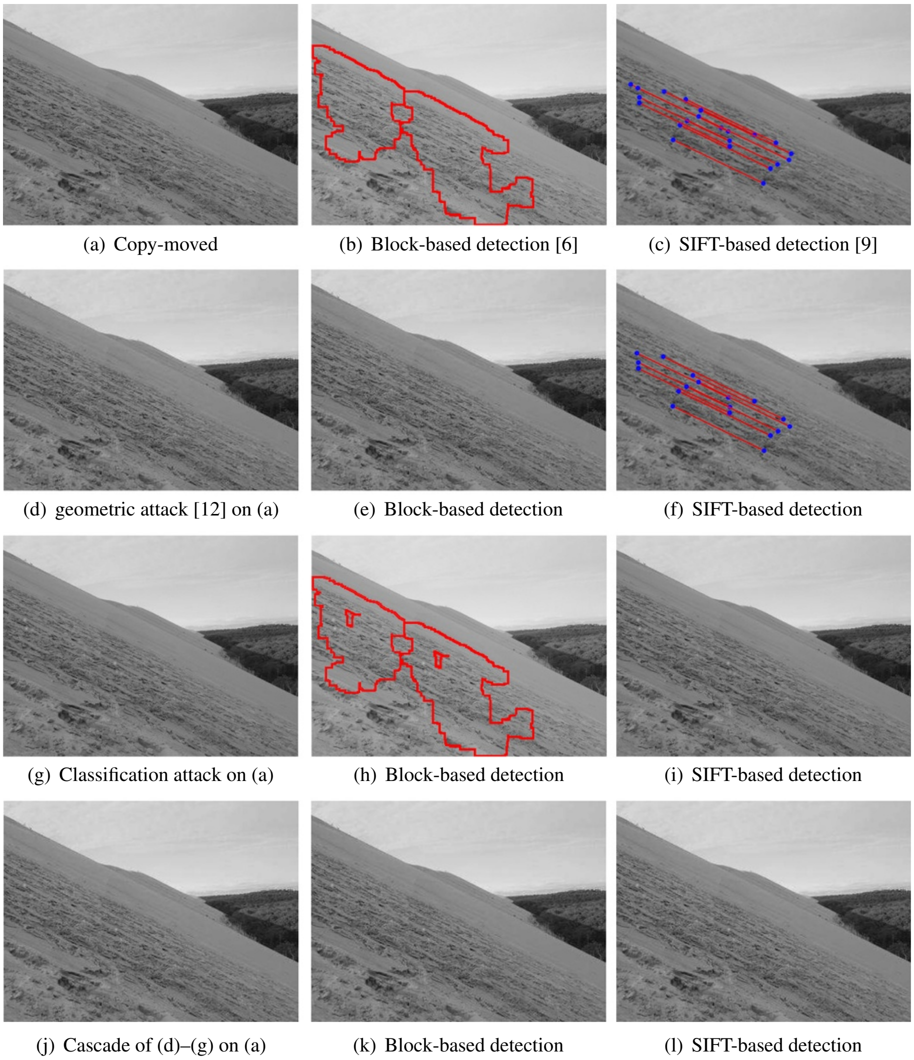
(a) Copy-moved
(b) Block-based detection [6]
(c) SIFT-based detection [9]
(d) geometric attack [12] on (a)
(e) Block-based detection
(f) SIFT-based detection
(g) Classification attack on (a)
(h) Block-based detection
(i) SIFT-based detection
(j) Cascade of (d)–(g) on (a)
(k) Block-based detection
(l) SIFT-based detection

Figure 14 Block-based copy-move detector [6] vs SIFT-based copy-move detector [9]. (a, b, c) On copy-move forged image, **(d, e, f)** following the geometric attack of [12], **(g, h, i)** following the classification-based attack, and **(j, k, l)** following the cascade of the two attacks.

The interpretation of such results is straightforward: SIFT features have been devised in such a way to be robust against geometric manipulations, which obviously fail at impairing a SIFT-based copy-move detector. On the other hand, the classification-based attack is designed to preserve the local visual quality of the image. Therefore, the artifacts introduced into the image do not alter the features analyzed by the block-based detector. As a consequence, the two different counter-forensic schemes (and possibly others) should cooperate with each other in order to be effective against a wider spectrum of detection techniques.

6 Conclusions

In this paper, we presented a counter-forensic scheme to counter a SIFT-based copy-move detector. The goal is to remove SIFT keypoints with the lowest possible impact on visual quality. To do so, we first classified SIFT keypoints depending on the histogram of their neighborhood. We then used attacks specifically tailored to each class. Results were better than those obtained by always using the same attack regardless of keypoint's properties. The proposed scheme was applied to a realistic copy-move scenario and succeeded in disabling a state-of-the-art SIFT-based detector. Several aspects could be further investigated, the most interesting of which is the injection of fake keypoints into the cleaned image. In fact, an image that does not contain SIFT keypoints (or very few of them) is suspicious: such absence could be taken as a clue of tampering, thus leading to a counter detector with a very straightforward implementation. As a matter of fact, in a copy-move scenario, the side effect of the classification-based attack tends to be less noticeable, mainly for two reasons: (1) only half of the keypoints are removed from each patch, and (2) some keypoints are actually not removed but altered in such a way that their previous match is canceled. Nonetheless, our attack could greatly benefit from an additional module introducing plausible fake keypoints and triggering false positives during the SIFT detection. Moreover, it could be useful to study more in depth the interactions between the counter-measures against SIFT-based and block-based copy-move detectors. Finally, it would also be interesting to apply our attack to a content-based image retrieval scenario in order to assess its effectiveness against SIFT-based search engines.

Endnote

[a]The UCID database can be freely downloaded from http://homepages.lboro.ac.uk/~cogs/datasets/ucid/ucid.html

Competing interests

The authors declare that they have no competing interests.

Authors' information

IA received the Laurea degree in Computer Engineering in 2006 and the PhD degree in Computer Engineering, Multimedia and Telecommunication in 2011, both from the University of Florence, Italy. She was a visiting scholar at Binghamton University, Binghamton, NY, USA in 2010. Currently, she is a postdoctoral researcher at the Image and Communication Lab at the Media Integration and Communication Center, University of Florence. Her main research interests focus on multimedia forensics and image processing.

MB graduated in Electronic Engineering at the University of Florence in 1991, where he received the PhD in Informatics and Telecommunications in 1995. He is currently working as associate professor at the University of Siena. During the last decade, his activity has focused on digital image processing and information security, with particular reference to the application of image processing techniques to copyright protection (digital watermarking) and multimedia forensics. Lately, he has been studying the possibility of processing signals that has been previously encrypted without decrypting them. He led several national and international research projects on these subjects. He is author of about 270 papers and holds four patents in the field of digital watermarking and document protection. He is co-author of the book *Watermarking Systems Engineering* (Dekker, 2004). In 2008, he was the recipient of the *IEEE Signal Processing Magazine* best column award. In 2011, he was the recipient of the IEEE Geoscience and Remote Sensing Society Transactions Prize Paper award. He has been the chairman of the IEEE Multimedia Signal Processing Workshop (Siena, 2004) and the chairman of the International Workshop on Digital Watermarking (Siena, 2005). He has been the founding editor in chief of the *EURASIP Journal on Information Security*. He currently serves as associate editor of the *IEEE Transactions on Circuits and System for Video Technology* and the *IEEE Transactions on Information Forensics and Security*. From 2010 to 2011, MB was the chairman of the IEEE Information Forensic and Security Technical Committee of the Signal Processing Society. He has been a member of the IEEE Multimedia Signal Processing Technical Committee and the conference board of the IEEE Signal Processing Society. He has been appointed distinguished lecturer of the IEEE Signal Processing Society for the years 2012-2013. MB is a fellow of the IEEE.

RC received the Laurea degree in Electronic Engineering in 1997 and the PhD degree in Computer Science and Telecommunication in 2001, both from the University of Florence, Italy. Currently, he is an assistant professor at the Media Integration and Communication Center of the University of Florence. His main research activities, witnessed by several publications, include digital image processing, image and video digital watermarking, multimedia forensics.

AC graduated in Telecommunications at the University of Siena in 2008. Currently, he is a PhD student at the Department of Information Engineering and Mathematical Science of the University of Siena. His research activity is focused on both image forensics and counter-forensics.

Acknowledgements

This work was partially supported by the REWIND Project, funded by the Future and Emerging Technologies (FET) programme within the 7FP of the EC under grant 268478.

Author details

Media Integration and Communication Center, University of Florence, Florence 50121, Italy. [2]Department of Information Engineering and Mathematical Sciences, University of Siena, Siena 53100, Italy.

Received: 29 October 2012 Accepted: 27 March 2013
Published: 15 April 2013

References

1. R Böhme, M Kirchner, in *Digital Image Forensics*, ed. by HT Sencar, N Memon. Counter-forensics attacking image forensics (Springer, New York, 2013), pp. 327–366
2. G Cao, Y Zhao, R Ni, H Tian, in *Proceedings of the 12th ACM Workshop on Multimedia and Security*. Anti-forensics of contrast enhancement in digital images (ACM, New York, 2010), pp. 25–34
3. M Stamm, S Tjoa, W Lin, K Liu, in *2010 17th IEEE International Conference on Image Processing (ICIP)*. Undetectable image tampering through JPEG compression anti-forensics (IEEE, New York, 2010), pp. 2109–2112
4. M Kirchner, R Bohme, Hiding traces of resampling in digital images. IEEE. Trans. Inf. Forensics Secur. **3**(4), 582–592 (2008)

5. S Bayram, H Sencar, N Memon, in *IEEE Western New York Image Processing Workshop*. A survey of copy-move forgery detection techniques (IEEE, New York, 2008), pp. 538–542
6. A Fridrich, B Soukal, A Lukáš, in *Proceedings of Digital Forensic Research Workshop*. Detection of copy-move forgery in digital images (Citeseer, PSU, Centre County, 2003)
7. H Popescu, AC Farid, Exposing digital forgeries by detecting traces of resampling. IEEE. Trans. Signal Process. **53**(2), 758–767 (2005)
8. X Pan, S Lyu, Region duplication detection using image feature matching. IEEE. Trans. Inf. Forensics Secur. **5**(4), 857–867 (2010)
9. I Amerini, L Ballan, R Caldelli, A Del Bimbo, G Serra, A SIFT-based forensic method for copy move attack and transformation recovery. IEEE. Trans. Inf. Forensics Secur. **6**(3), 1099 –1110 (2011)
10. DG Lowe, Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
11. V Christlein, C Riess, J Jordan, C Riess, E Angelopoulou, An evaluation of popular copy-move forgery detection approaches. IEEE. Trans. Signal Process. **7**(6), 1841–1854 (2012)
12. S Nguyen, HC Katzenbeisser, in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Security of copy-move forgery detection techniques (IEEE, New York, 2011), pp. 1864 –1867
13. SC Hsu, CY Lu, CS Pei, in *Proceedings of the 17th ACM International Conference on Multimedia*. Secure and robust SIFT (ACM, New York, 2009), pp. 637–640
14. E Do, TT Kijak, T Furon, L Amsaleg, in *Proceedings of the International Conference on Multimedia*. Understanding the security and robustness of SIFT (ACM, New York, 2010), pp. 1195–1198
15. E Do, TT Kijak, T Furon, L Amsaleg, in *Proceedings of the 2nd ACM Workshop on Multimedia in forensics, Security and Intelligence*. Deluding image recognition in sift-based cbir systems (ACM, New York, 2010), pp. 7–12
16. E Do, TT Kijak, L Amsaleg, T Furon, in *37th International Conference on Acoustics, Speech, and Signal Processing, ICASSP'12*. Enlarging hacker's toolbox: deluding image recognition by attacking keypoint orientations (IEEE, New York, 2011)
17. Y Liu, D Zhang, G Lu, W Ma, A survey of content-based image retrieval with high-level semantics. Pattern Recognit. **40**, 262–282 (2007)
18. R Caldelli, I Amerini, L Ballan, G Serra, M Barni, A Costanzo, in *Proceedings of International Symposium on Communications, Control and Signal Processing (ISCCSP)*. On the effectiveness of local warping against SIFT-based copy-move detection (IEEE, New York, 2012)
19. YL Chang, JH Fan, KC Chang, Multi-modal gray-level histogram modeling and decomposition. Image Vis. Comput. **20**(3), 203–216 (2002)
20. D Zhang, G Lu, in *Proceedings of the International Conference on Neural Networks and Signal Processing, 2003*, vol. 2. Evaluation of similarity measurement for image retrieval (IEEE, New York, 2003), pp. 928–931
21. G Schaefer, M Stich, in *Storage and Retrieval Methods and Applications for Multimedia 2004, volume 5307 of Proceedings of SPIE*. UCID - An uncompressed colour image database (SPIE, Bellingham, 2004), pp. 472–480
22. A Vedaldi, B Fulkerson, VLFeat: an open and portable library of computer vision algorithms (2008). http://www.vlfeat.org/. Accessed 09 April 2013
23. A Vedaldi, An open implementation of the SIFT detector and descriptor (2007). Technical report 070012, UCLA CSD
24. R Hess, in *Proceedings of the International Conference on Multimedia*. An open-source SIFTLibrary (ACM, New York, 2010), pp. 1493–1496

doi:10.1186/1687-5281-2013-18
Cite this article as: Amerini *et al.*: Counter-forensics of SIFT-based copy-move detection by means of keypoint classification. *EURASIP Journal on Image and Video Processing* 2013 **2013**:18.

Submit your manuscript to a SpringerOpen journal and benefit from:

- ► Convenient online submission
- ► Rigorous peer review
- ► Immediate publication on acceptance
- ► Open access: articles freely available online
- ► High visibility within the field
- ► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com