EURASIP Journal on Information Security
a SpringerOpen Journal

RESEARCH Open Access

Removal and injection of keypoints for SIFT-based copy-move counter-forensics

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

Abstract

Recent studies exposed the weaknesses of scale-invariant feature transform (SIFT)-based analysis by removing keypoints without significantly deteriorating the visual quality of the counterfeited image. As a consequence, an attacker can leverage on such weaknesses to impair or directly bypass with alarming efficacy some applications that rely on SIFT. In this paper, we further investigate this topic by addressing the dual problem of keypoint removal, i.e., the injection of fake SIFT keypoints in an image whose authentic keypoints have been previously deleted. Our interest stemmed from the consideration that an image with *too few* keypoints is per se a clue of counterfeit, which can be used by the forensic analyst to reveal the removal attack. Therefore, we analyse five injection tools reducing the perceptibility of keypoint removal and compare them experimentally. The results are encouraging and show that injection is feasible without causing a successive detection at SIFT matching level. To demonstrate the practical effectiveness of our procedure, we apply the best performing tool to create a forensically undetectable copy-move forgery, whereby traces of keypoint removal are hidden by means of keypoint injection.

Keywords: Counter-forensics; SIFT; Keypoint injection; Keypoint removal

1 Introduction

Counterfeiting digital images by means of photo editing tools to alter the original meaning is becoming an immediate and easy practice. One of the most common ways of manipulating the semantic content of a picture is copy-move forgery, whereby a portion of the image is copied and pasted once or multiple times elsewhere into the same image. Image forensic literature offers several examples of detectors for such manipulation [1]; among them, the most recent and effective ones [2,3] are those based on scale-invariant feature transform (SIFT) [4]. The capability of SIFT to discover correspondences between similar visual content, in fact, allows forensic analysis to detect even very accurate and realistic copy-move forgeries.

Expectedly, a methodology that is so powerful has drawn the interest of counter-forensic research, where with the term *counter-forensics* we refer to the study of methods to counter-attack forensic techniques by concealing manipulation traces [5]. The actual reliability of forensic algorithms can only be estimated by considering what an attacker can try to do to invalidate them. Furthermore, since SIFT is a powerful instrument to recognise and retrieve objects, an analysis on SIFT security becomes very important also in the case of content-based image retrieval (CBIR) [6] systems, in order to assess if an attacker is able or not to succeed in deluding the image recognition process. The first work in this sense is the one by Hsu et al. [7], in which first the impact of simple attacks is analysed and then a method to strengthen SIFT features (or keypoints) is proposed. Following this work, Do et al. [8-10] focused on a SIFT-based CBIR scenario and devised a number of interesting attacks. Caldelli et al. [11] are the first to address the complete removal of the keypoints by means of an attack based on local warping techniques derived from image watermarking [12]. Recently, Amerini et al. [13] proposed a keypoint removal scheme based on the classification of the neighbourhood of each keypoint, followed by an *ad hoc* attack for each class. All these studies have demonstrated that devising methods to attack SIFT features is not a trivial task: SIFT features, in fact, are not only robust to several non-malicious processing but also to attempts of tampering. As a consequence, most attacks, though succeeding in

*Correspondence: andreacos82@gmail.com
[1] Department of Information Engineering and Mathematical Sciences, University of Siena, Via Roma, 56, Siena 53100, Italy
Full list of author information is available at the end of the article

erasing keypoints, pay a high cost in terms of visual quality degradation.

Given that, anyway, there is another issue to be taken into account when performing keypoint deletion: an image that does not contain SIFT keypoints (or very few of them) is suspicious; such absence, especially in textured areas, could be taken as a clue of tampering, thus potentially allowing to devise forensic detectors revealing the manipulation. Although currently there exists no other detector than visual inspection, a smarter attack could greatly benefit from an additional module introducing (or *injecting*) plausible fake keypoints which could trigger false positives during the SIFT match detection. Reinserted keypoints should ideally appear in a neighbourhood of the original spatial locations, but, at the same time, their SIFT descriptors should be as far as possible from the original ones in the SIFT space. In addition to that, the number of inserted keypoints should be as high as possible and their spatial distribution should comply with the underlying image content (a huge number of thickened keypoints could be questionable as well). This topic is crucial in a copy-move forgery detection scenario where portions of an image are to be considered.

In this paper, we investigate the injection of fake SIFT keypoints in an image whose authentic keypoints were previously removed while still avoiding matching in the SIFT domain. With this aim, we analyse five different algorithms. The fundamental idea of our study is to highlight the SIFT security issue deriving from keypoint injection after a previous removal, to provide some instruments to perform this action and, finally, to present an analysis on some initial results.

The paper is organised as follows. We first briefly review the procedure used to remove SIFT keypoint and its counter-forensic application to copy-move detection. Secondly, we give a glance of our idea about keypoint injection and describe the tools we use to perform it. Then, we experimentally validate the effectiveness of the tools; the most reliable tool is used to produce a forensically undetectable copy-move forgery. Such a forgery, in fact, is still capable of bypassing a SIFT-based state-of-the-art detector without exhibiting traces of keypoint removal. We conclude the paper by outlining some directions for future research.

2 SIFT-based copy-move detection

In the following, we will assume that the reader is already accustomed to the theory underlying the SIFT algorithm, for which we refer to [14] and [4]. SIFT allows to model complex objects or scenes by a collection of multiscale distinctive local features that are invariant to scale and rotation and robust, to a variable extent, to affine distortions, changes in illumination, changes of 3D viewpoint, cluttering, occlusions, and noise addition. These features are obtained from the neighbourhoods of salient points referred to as keypoints; in general, SIFT provides a high amount of keypoints, even if dependent on the image's content and size, densely distributed across the image, at a low computational cost. To each keypoint is associated a descriptor, i.e., a compact vector capturing the properties of the keypoint neighbourhood, whose high discriminative power allows for robust and reliable matching between similar images, thus explaining the widespread adoption of SIFT in fields such as image retrieval, image clustering, or object recognition.

Generally, the SIFT operator is applied to two images. In the case of copy-move detection, it is instead applied to one image only since the copied part is within the same image. Expectedly, the descriptors extracted from a cloned region are quite similar to those of the source, thus making possible to discover the manipulation by matching keypoints. During this process, one can also retrieve information about the geometric transformation that has been applied to the cloned region.

The algorithm we aim to counter [3] is based on the above rationale. In a nutshell, it works as follows (see Figure 1). Given an image I, the method first extracts the keypoints $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and their descriptors $D = \{f_1, \ldots, f_n\}$. Then, the best candidate match for each keypoint $\mathbf{x}_i$ is found by identifying its nearest neighbour among the other $n-1$ keypoints, i.e., the keypoint with the minimum Euclidean distance descriptor.

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

Although this set of linked, isolated keypoints already provides a rough idea of the presence of cloned areas, a clustering procedure is run for improved accuracy. To assess the existence of cloned areas, an *agglomerative hierarchical clustering* [15] is carried out: (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. The procedure is repeated until no more pairs can be merged. Upon agglomerating, if two or more clusters are detected with at least 4 pairs of matched points linking each other, then the corresponding regions are deemed cloned.

3 SIFT keypoint removal

We now briefly describe the method to remove SIFT keypoints that we presented in [13]. Such a method, called classification-based attack (briefly CLBA), adopts a strategy based on two observations. The first observation is that it is possible to discriminate between SIFT keypoints according to some of their local properties (in [13], the

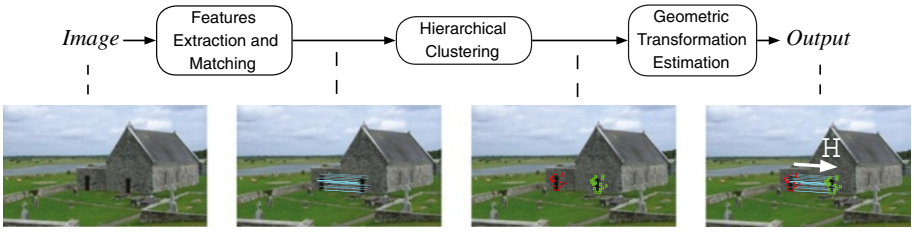

Figure 1 SIFT-based copy-move detection. Work-flow of the detector of [3] impaired by the counter-forensic attack of [13].

first-order statistics of surrounding regions in the pixel domain). The second observation is that each class of keypoints reacts differently to counter-forensic attacks. As a consequence, the use of attacks specifically tailored to a class ensures a reduced impact on the counterfeited image's quality.

More specifically, classification is basically done by resorting to a histogram description of a squared neighbourhood around every keypoint; on the basis of the histogram shape, three classes are defined according to the number of modes: unimodal, bimodal, and multimodal. Each class of keypoints is then removed by means of a dedicated attack in an iterative fashion that halts when a condition, such as the desired percentage of removed keypoints or the maximum number of allowed iterations, is met. Two main reasons suggest the use of an iterative procedure: (1) some keypoints are easier to remove than others; for the sake of visual quality of the forgery, the former are attacked with less strength than the latter. Therefore, an iterative approach naturally allows to intensify the strength of the attack as more robust keypoints keep surviving; (2) changes in pixel values, in the attempt to remove a keypoint, may accidentally generate a new keypoint in the proximity. This phenomenon occurs in the neighbourhoods of those candidate keypoints that were discarded because they are barely below the SIFT thresholds. CLBA does not keep track of this different categories of keypoints but rather attempts to remove them in the successive iterations.

For the first part (usually one third of the iterations), all the keypoints are attacked with the Smoothing, and for the second part, they are attacked with Collage and removal with minimum distortion (RMD) [9]. The Smoothing attack reduces the population of keypoints without a significant loss of quality. The keypoints that survive to this first round require more powerful countermeasures, i.e., Collage (unimodal and multimodal classes) and RMD (bimodal class). In the following, we briefly review each attack taken in account.

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. The strength 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 removal rate and the overall visual quality after the attack. This attack has also been used in [9].

The Collage attack is a variant of the method used in [7]. In general, it consists in the substitution of an authentic image patch with another patch of the same size but with different properties. In our case, the new patch should obviously not contain SIFT features and should be as similar as possible to the original one, according to the some similarity criteria. We chose to measure such similarity by means of the histogram intersection distance [16], which has been widely used in the past in image retrieval applications.

Let then P_{orig} and P_{min} be respectively the authentic patch containing keypoints and its most similar keypoint-free patch stored in a large database (i.e., the patch whose histogram is at minimum distance); to avoid visible artefacts along the borders, we do not reinsert P_{min} directly into the original image. Instead, we reinsert the following linear combination:

$$P_{\text{new}} = W \times P_{\text{orig}} + (1 - W) \times P_{\text{min}}, \tag{1}$$

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.

The third attack is the RMD attack proposed by Do et al. in [9]. The idea behind this technique is to calculate a small patch ϵ added to the neighbourhood of a keypoint which allows its removal. The coefficients of ϵ are chosen in such a way to reduce the contrast around the keypoint computed at the DoG level, thus invalidating the check performed by the SIFT algorithm on all potential keypoints. Moreover, it is requested that the coefficients locally introduce the minimum visual distortion, and differently from the original version of the algorithm, we used the same weighting window of Equation 1 to replace the original neighbourhoods with the new patch.

The performance in successfully removing keypoints is evaluated by means of the keypoint removal rate (KRR) metric, which is defined as follows:

$$\mathrm{KRR} = \left(1 - \frac{\text{Keypoints detected after attack}}{\text{Keypoints detected before attack}}\right) \times 100. \tag{2}$$

The experimental analysis carried out in [13] shows that CLBA outperforms the state-of-the-art of class-unaware keypoint removal attacks both in terms of KRR and of the impact on the quality of the counterfeited image.

3.1 Application to copy-move

In [13], we applied CLBA to the image forensic scenario to impair a state-of-the-art SIFT-based copy-move detector presented in [3]. Without loss of generality, we assumed to work on copy-move forgeries consisting of two cloned areas. In this scenario, it is possible to conceal the manipulation by attacking only the matching keypoints revealed by the copy-move detector. Since to remove a match suffices to delete only one member, keypoint removal can be distributed over the two cloned regions. At each iteration, only one keypoint of each match is manipulated. Let N_m be the number of matches between the two copy-moved regions R_1 and R_2, revealed by the forensic detector: first, $\frac{N_m}{2}$ matches are randomly picked, and the corresponding keypoints in R_1 are erased; then, the same thing is done for the remaining $\frac{N_m}{2}$ matches, by attacking only the corresponding keypoints in R_2. A difference with the standard attack is that, now, the effectiveness is not measured in terms of KRR, but rather in terms of match removal rate (MRR):

$$\mathrm{MRR} = \left(1 - \frac{\text{Matches detected after attack}}{\text{Matches detected before attack}}\right) \times 100, \tag{3}$$

where obviously only matches across cloned regions are considered. In practical terms, the halting conditions of the attack are controlled by a target match removal rate and a maximum number of iterations.

4 SIFT keypoint injection

We considered five possible attacks to introduce fake keypoints. Three of which in particular are based on adaptive image enhancement algorithms already known in the literature but never used before in the context of SIFT-countering. The rationale behind re-purposing locally adaptive image enhancement for keypoint injection is the following. It has been observed that smoothing techniques perform well in the removal since they reduce image details: therefore, we may argue that enhancement techniques, which in turn exalt details, should conversely introduce new keypoints. Moreover, since traditional techniques applied to the whole image (e.g., sharpening, global contrast enhancement) generate visually unpleasant images, more sophisticated solutions should be employed so that the resulting quality is comparable to that of the authentic image. The experimental results we obtained confirm the validity of both intuitions. In addition to the image enhancement tools, we also considered two attacks representing the symmetrical versions of SIFT-countering methods already known in the literature [9]: Gaussian smoothing and forging with minimum distortion (FMD).

The framework that we will use to evaluate the capability of injection is the same for all the methods. Therefore, for the sake of clarity, we introduce its working principles here in accordance with the schematization of Figure 2.

In the sequel, the injection attack is applied to a counter-forensically treated (or *cleaned*) image I_{rem}, whose keypoints have been removed by means of a certain attack, such as, for example, CLBA. Unlike removal, during keypoint injection, the image is processed full-frame, so this fact has a negative impact on the visual quality of the entire image (e.g., flat areas which did not contain keypoints originally). For this reason, in an 8×8 neighbourhood of each keypoint we mix this image, let us call it pre-injected image, with the original one I, in a way that is similar to how we did in Equation 1 for the removal procedure. The obtained patches are then substituted onto the cleaned image I_{rem} producing the final injected image I_{inj} which now shows a better visual quality.

As a final step, it is necessary for the attacker to check how many injected keypoints in the image are really valid. An injected keypoint is deemed as valid when, first of all, it is located in a textured area and is spatially distributed with respect to the others and, above all, has a SIFT descriptor which is sufficiently different from its original homologue not to evidence a match. It is worth to point out that it is not so crucial to check that the new injected keypoint is or is not in the same spatial position (x-y coordinates) with respect to the original one. Thus, on this basis, we perform a matching detection between the original image I and the injected one I_{inj}. Ideally, it would be desirable not to obtain matches between the two images, though having in the injected image I_{inj} a plausible amount of well distributed keypoints. In practice, we refine the injected image as follows: first, we individuate the new keypoints presenting a correct match with their homologue (wrong matches are instead welcome); second, we discard them by inserted back into I_{inj} a corresponding patch (16×16 pixels which is the computational window of the SIFT descriptor) of the cleaned image I_{rem}. The background idea is to primarily avoid a SIFT match at the expense of the loss of an injected keypoint. At the end of

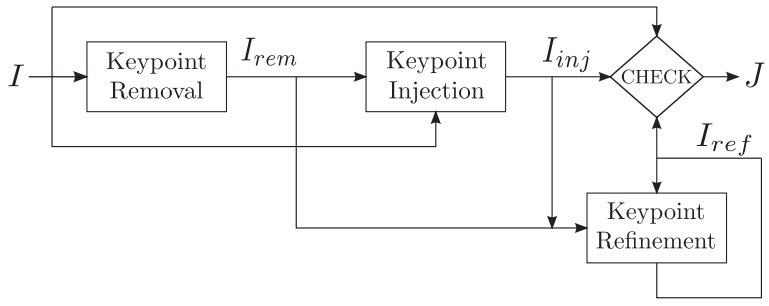

Figure 2 General framework for keypoint injection.

the refinement (consisting of one loop in this work), the final attacked image J is obtained.

The attacks used to forge fake keypoints rely on five different algorithms, which are briefly described in the following subsections.

4.1 Contrast-limited adaptive histogram equalisation

Global contrast enhancement techniques assume that the distribution of gray scale pixel values is uniform over all the areas of an image. When this assumption does not hold, performances of global methods are poor and the enhanced images are visually unpleasant. Contrast-limited adaptive histogram equalisation (CLAHE) [17] tackles with this problem in two ways: it adapts to the local properties of the regions of an image and limits the contrast differences across them. In a nutshell, the algorithm proceeds as follows (see [17] for details). First, the $M \times N$ gray scale image I is divided into non-overlapping tiles, and the histogram of each tile is computed. Then, a clipping limit β for the contrast enhancement is obtained as in Equation 4:

$$\beta = \frac{MN}{L}\left(1 + \frac{\alpha}{100}\left(s_{\max} - 1\right)\right), \tag{4}$$

where L is the number of histogram bins, $\alpha \geq 0$ is the *clipping factor*, and $s_{\max}$ is the slope of the transfer function mapping the contrast from its input value to its output value; if $s_{\max} = 1$, then no enhancement is performed, while larger values (usually up to 4) will result into more visible enhancements. Next, each histogram is clipped in such a way that its height is limited by β. At this point, it is necessary to remap the clipped values to the entire intensity range, that is, to renormalise the histogram of the processed image to its original area. This task can be carried out in several ways, the most common of which consists in redistributing the clipped pixels uniformly in all the bins of the histogram of the whole image.

4.2 Brightness preserving dynamic fuzzy histogram equalisation

Brightness preserving dynamic fuzzy histogram equalisation (BPDFHE) is a method to enhance the contrast of an image while preserving its mean brightness and thus the perceived subjective quality [18]. Similar to other contrast enhancement techniques, it proposes to divide the image histogram into segments, which are then independently equalised. Partitioning, however, is not performed on the normal histogram but rather on its fuzzy counterpart, whereby a pixel may belong to some degree to more than one of the bins, in accordance with a fuzzy membership function. Such a histogram, in fact, is generally smoother, with no missing levels or abrupt fluctuations, thus allowing a more accurate segmentation. The pixel membership functions can be designed in different fashions depending on the application. The most common functions are triangular and Gaussian ones. The algorithm proceeds as follows (see [18] for details):

1. The fuzzy histogram $\widetilde{H}(k)$, $k = [0, L]$ is computed by assigning to each bin k the number of pixels whose value is *around* k, in accordance with the chosen membership function.
2. The local maxima $\{m_1, m_2, \ldots, m_n\}$ are computed and used to define the segments of the histogram: $S = \{\,[0, m_1 - 1], [m_1, m_2 - 1], \ldots, [m_n, L]\,\}$.
3. Each segment is equalized by means of a technique depending on the number of pixels belonging to the partition.
4. In order to cope with the alterations that may have been introduced, the resulting brightness is normalised to match the original brightness.

4.3 Anisotropic diffusion

Two-dimensional anisotropic diffusion (2D-AD) is a method to enhance images by preserving the perceptual quality of semantically relevant parts, such as straight lines, edges, and geometric shapes [19]. In principle, it is a generalisation of the scale-space transform, whereby an image I is iteratively convolved with a non-linear

smoothing filter, which is adapted to the local content to generate progressively more blurred versions of I (theoretical details are left to [20] and [21]). In other words, the key idea behind 2D-AD is to keep image structures intact by smoothing only the area around them.

The filter model allowing anisotropic diffusion is derived from well-known operators used to extract image details. Let $I_\sigma = I * G_\sigma$ be the convolution of an image I with a Gaussian kernel ($\sigma > 0$); then, the gradient ∇I_σ can be employed to highlight structures like the edges of I. Since the gradient does not always perform satisfactorily, a more effective operator is derived from it.

Let $J(\nabla I_\sigma) = \nabla I_\sigma \nabla I_\sigma^T$; then, $J_\rho(\nabla I_\sigma) = J(\nabla I_\sigma) * G_\rho$, that is, the convolution with a Gaussian kernel ($\rho > \sigma$), is called *tensor operator*, and it can be used to effectively highlight flow-like, T-shaped or Y-shaped structures [20]. The eigenvectors $\{w_1, w_2\}$ of J_ρ give indications on local orientations as well as the corresponding eigenvalues (μ_1, μ_2) on the local contrast along these directions. The 2×2 *diffusion tensor* D that permits to perform the anisotropic diffusion is created in such a way that its eigenvectors are the same with J_ρ, and its eigenvalues (λ_1, λ_2) are

$$\lambda_1 = c_1$$
$$\lambda_2 = \begin{cases} c_1 & \text{if } \mu_1 = \mu_2 \\ c_1 + (1 - c_1) \exp\left[\frac{-c_2}{(\mu_1 - \mu_2)^2}\right] & \text{otherwise,} \end{cases} \quad (5)$$

where $c_1 \in (0, 1)$ and $c_2 > 0$. The elements of D are derived from (λ_1, λ_2). The result is an operator that can steer the enhancement according to the direction of flow-like structures in the image. By resorting to D, it is possible to efficiently compute blurred versions of $I(x, t)$ as numerical solutions of Equation 6, where $t \geq 0$ is called *diffusion time*:

$$\frac{\partial I}{\partial t} = \nabla \cdot (D \nabla I). \quad (6)$$

In practice, the algorithm proceeds as follows: given $I = I(x, 0)$, first $J_\rho(\nabla I_\sigma)$ is computed, and D is derived with Equation 5; then, $I(x, 1)$ is obtained with Equation 6. Starting from $I(x, 1)$, the process is repeated until a specified number of iterations is reached (i.e., $t \leq t_{\max}$).

4.4 FMD and Gaussian smoothing

In a nutshell, FMD is the symmetrical version of the RMD attack used to remove keypoints [9]. Given a location (x, y) and a scale σ at which a keypoint should be introduced, this time, the patch ϵ that needs to be added to the neighbourhood of (x, y) is computed by solving the following problem:

$$\epsilon = \underset{\epsilon : D'(\mathbf{x}) = D(\mathbf{x}) - \delta}{\operatorname{argmin}} \frac{1}{2} \|\epsilon\|^2. \quad (7)$$

The parameter δ still controls the intensity of the attack: $D(x, y, \sigma)$ is raised by $|\delta|$ in such a way that the altered value $D'(x, y, \sigma)$ in now greater than the contrast threshold.

Finally, it has been experimentally observed that Gaussian smoothing can introduce keypoints into an image if the filter's width is large enough. In practice, good results can be obtained by letting $h = 3$ and $\sigma = 2$.

Among all attacks, only FMD allows to decide exactly the location (x, y) where the fake keypoint is inserted. As a consequence, it becomes necessary a criterion to choose such locations. For this implementation, we have chosen the locations of structures such as edges, T shapes and Y shapes, detected by the tensor matrix D also used by 2D-AD. The reason why we use such a large amount of candidate locations resides in the characteristics of the FMD. The artefacts it introduces are not visually acceptable if the attack is too intense, thus forcing us to set the strength to its minimum, i.e., $\delta = 1$. By doing so, however, the attack cannot ensure a positive outcome of the injection. For this reason, we repeat the injection for all the locations, and following each attempt, we perform a SIFT check to verify whether a new keypoint has been injected. If this is the case, we do not inject other keypoints in the 8×8 neighbourhood of the new keypoint, in avoiding to accidentally remove it.

5 Experimental results

In this section, we first evaluate the performance of the injection methods in general, in terms of both the injection effectiveness and the impact on the visual quality of the forgery. Then, we apply one of the injection tools to a copy-move detection scenario in which we hide the keypoint removal preceding injection.

The effectiveness of an injection attack is measured with the keypoint injection rate (KIR) as follows:

$$\text{KIR} = \left(\frac{\text{New keypoints following injection}}{\text{Keypoints before removal}} \right) \times 100, \quad (8)$$

where the numerator takes into account only the injected keypoints and not those left in the image by non-perfect removal.

A second important metric is the number of fake keypoints correctly matching with the authentic ones (obviously, the less, the better). We consider a SIFT match correct if it links two keypoints located in the same spatial position or, at most, within an 8×8 neighbourhood. The final quality of the forged image is evaluated both globally and locally by means of peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) index [22].

Concerning the experimental setup, keypoints are detected by means of the VLFeat SIFT implementation

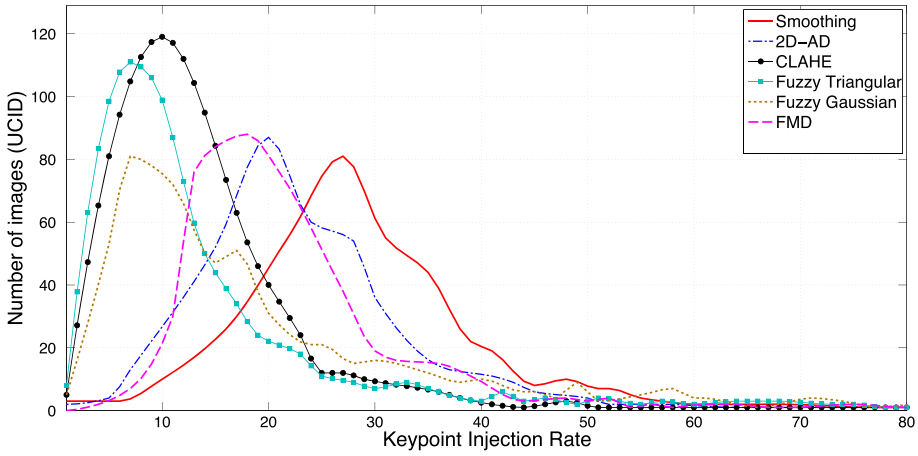

Figure 3 KIR envelope for each of the methods.

[23]. To obtain descriptors that are as close as possible to the original implementation by Lowe [4], we set SIFT thresholds as follows: edgeThreshold = 10, peakThreshold = 4; matching is performed with nearest neighbour (threshold fixed to 0.8), as suggested in [4]. The parameters for the removal stage are the same with those of [13]: the removal support, the target removal rate, and the number of iterations are set respectively to 8, 100%, and 40. Finally, the size of the injection support is set to $M = 8$. The rest of the parameters of each technique have been set to the values indicated in the reference papers.

5.1 Effectiveness of the injection attacks

All the injection attacks are evaluated on the UCID image database [24], consisting of 1,338 uncompressed (TIFF) colour images, whose size is either 384×512 or 512×384 pixels. All such images have been previously attacked with CLBA.

For each injection tool, we manipulated all the cleaned images and organised the corresponding KIRs into histograms, whose envelopes are shown in Figure 3.

The performance of most of the methods are quite similar, with Gaussian smoothing (average KIR = 27.9%), 2D-AD (23.4%) and FMD (17%) being superior to CLAHE (14%) and BPDFHE (either membership function, 11%). It is worth noting that the methods allow to introduce a larger number of keypoints with non-default parameters if one is willing to trade off with a reduced image quality. In Table 1, for each attack, we report the average PSNR and SSIM between the original and the injected patches.

We used the 27 images with a removal rate of at least 98% to create the graphs of Figures 4 and 5, showing the relationships between keypoints preceding and following the various manipulations. In particular, for each plot, the upper curve (red star marker) indicates the original number of keypoints, the lower curve (blue square marker) indicates the number of keypoints following the removal attack, and the curve between them represents the number of keypoints following injection of a selection of methods: CLAHE, Gaussian smoothing, 2D-AD, and FMD.

We then investigated a potential side effect of keypoint injection, i.e., the introduction of keypoints correctly matching with their homologues in the authentic image. In Figure 6, we show the distributions of correct matches for authentic images (left), following the removal-injection for Smoothing (centre) and 2D-AD (right). The number of matches is significantly lower in the injected images (results are similar for the other attacks, which we omit for sake of brevity). The two distributions following injection are slightly different because, following the refinement of the injected image, effects along borders may generate new keypoints. However, it is worth noting that, in most of the cases, the matches are caused by non-perfect removal rather than by the injection procedure. Therefore, a possible solution to further reduce their population could consist in increasing

Table 1 Impact of keypoint injection on visual quality

	Smoothing	2D-AD	CLAHE	Fuzzy triangular	Fuzzy Gaussian	FMD
PSNR	26.01	32.51	31.22	35.40	29.92	29.01
SSIM	0.810	0.898	0.964	0.959	0.953	0.937

Average full-frame PSNR and SSIM for the injected UCID data set.

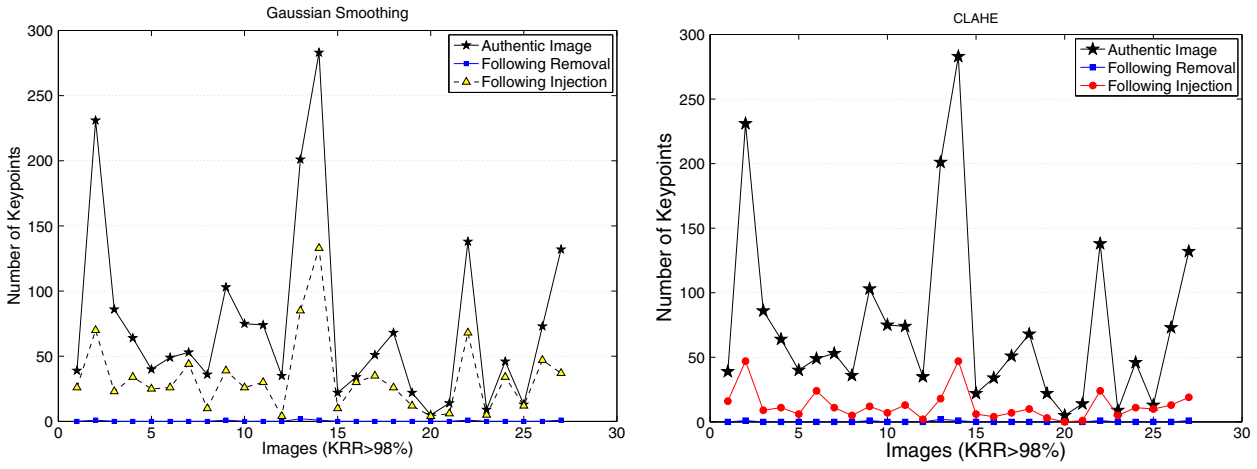

Figure 4 Keypoints per image for the removal-injection procedure. Left: Gaussian smoothing and and right: CLAHE.

the strength of the removal attack at the cost of image quality.

We conclude this section with some pointers to the complexity of the injection attacks. In general, injection is not demanding in terms of computational resources. Since the full-frame attacks do not permit to decide the spatial location of the injected keypoints and thus do not require any particular iteration or check, their time complexity is very low and never exceeds 15 s. FMD represents the only exception and requires about 60 s to produce the injected image. This behaviour can be explained with the rather high amount of iterations (several thousands) required by FMD to attempt the injection in all the candidate locations and to verify its outcome by means of a SIFT check. All tests have been performed on MathWorks Matlab running on a desktop configuration with 2 GHz dual-core processor, 4 GB RAM, 32-bit Windows OS.

5.1.1 Examples of keypoint injection

In Figure 7, a detailed example of the removal-injection procedure is given. From left to right, the top row shows the keypoints of authentic (blue squares), cleaned (perfect removal), and injected (red circles) images. The bottom row shows the authentic image's matches with cleaned (left) and injected (right) images. The three following aspects are worth the attention:

- *Concerning keypoints.* The cleaned image does not contain any keypoint, which is quite suspicious per se, being the image content that is very textured, as clearly evidenced by the authentic distribution of keypoints. On the contrary, the injected image looks more natural: the absence of keypoints over the clouds, which is an almost flat area, is not so strange. The original number of keypoints was 73; the injected amount is 35.

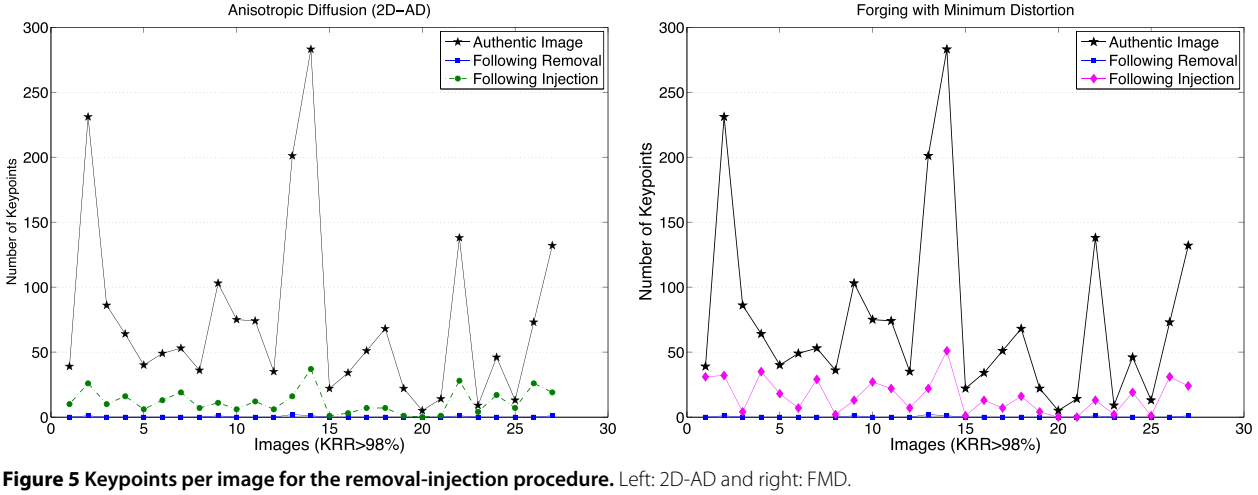

Figure 5 Keypoints per image for the removal-injection procedure. Left: 2D-AD and right: FMD.

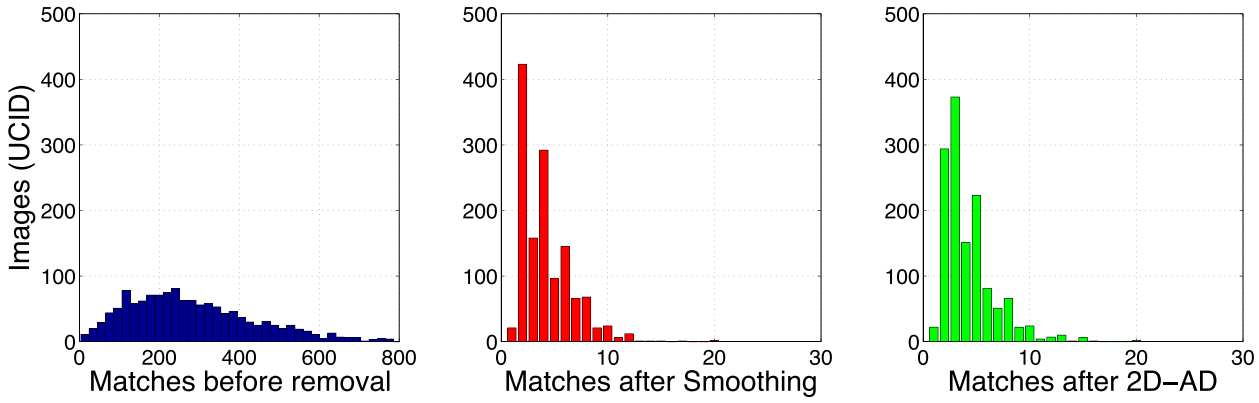

Figure 6 Histograms of correct matches. Left: before removal, middle: following injection by means of Smoothing attack, and right: following injection by means of 2D-AD (similar behaviour for omitted methods).

- *Concerning matches*. Obviously, the cleaned image does not produce any match. Although the injected image has 12 matches (Figure 7, bottom right), only the one on the left spire is correct (i.e., it falls inside the 8×8 neighbourhood of its authentic homologue). This is a very interesting outcome, confirming that injection can support removal not only by concealing keypoint removal traces but also by further misleading any analysis that is based on matching.
- *Concerning quality*. The injection procedure does not deteriorate excessively the manipulated image. With respect to the authentic image, the following quality metrics are obtained: the average PSNR and SSIM computed on all the attacked neighbourhoods are 32.8 dB and 0.9671, respectively, full frame PSNR is 38.9 dB, and full-frame SSIM is 0.9941. In terms of full-frame PSNR, the injection causes a small loss of 2.5 dB with respect to the cleaned image.

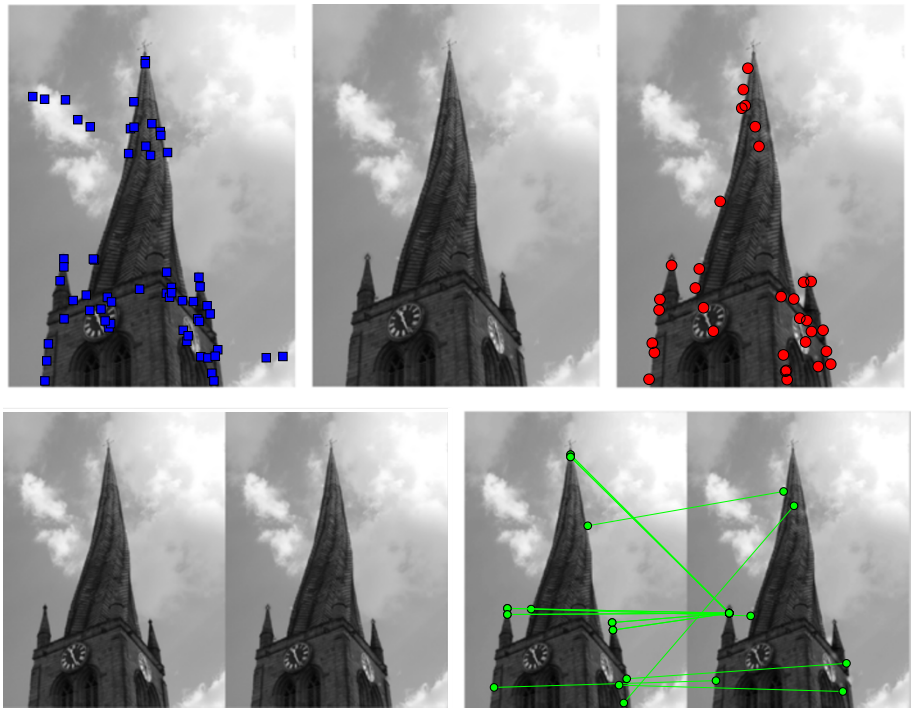

Figure 7 Example of removal-injection procedure. From top left to bottom right: authentic image, cleaned image, injected image, matches between authentic and cleaned images, and matches between authentic and injected images.

Figure 8 Example of copy-move removal-injection attack (1). The number of keypoints and of SIFT-matches when copy-move attack is performed without keypoint removal, with keypoint removal, and with keypoint removal-injection.

5.2 Application of keypoint removal-injection to copy-move detection scenario

In this section, we applied one of the above tools, namely the 2D-AD, to a copy-move detection forensic scenario. More specifically, we first impaired a state-of-the-art SIFT-based copy-move forgery detector (briefly CMFD) [3] by means of the CLBA attack [13]; then, we hid potential traces of keypoint removal using the aforementioned injection procedure.

In copy-move counter-forensics, the aim of an attacker is to avoid that a specialised forensic tool can detect SIFT-matched keypoints by linking the cloned patch to its source; while pursuing such a goal, it is not advisable to delete all the keypoints of the cloned area(s) if the imperceptibility of the attack is to be preserved. For the sake of simplicity, we will assume, without loss of generality, to deal with two copy-moved patches (i.e., no multiple copies). Therefore, our objective is to attack only the source and/or destination patches, rather than the whole image.

In this experiment, we decided to remove and subsequently inject keypoints in only one of the two patches, i.e., the destination one, by keeping the other unaltered. The goal of the experiment is to show the effectiveness of the keypoint removal-injection procedure in fooling the chosen copy-move forgery detector. Such a CMFD, in particular, requires a number of four matches to claim that an image has been tampered with.

Figure 8 shows the results we obtained on a test image, whereby the window of the left side of the church has been cloned onto the central façade. The first row shows the SIFT keypoints for the copy-moved (left), cleaned (center), and injected (right) images. The second row shows the corresponding output of the CMFD for the above images. It can be observed that the CMFD is able to recognise the forgery (bottom left), but it is fooled when keypoint removal is applied (bottom centre) since no matches have been found. However, the reduction in the number of keypoints within the destination patch (top centre) is visually noticeable with respect to the authentic image (top left). Following keypoint injection (bottom right), no matches appear between the cloned patches, so the duplication is still not recognised, though three new keypoints have been reinserted in the destination patch. In Table 2, a detailed numerical evaluation is proposed.

In Figure 9, details regarding the cloned areas of another image is shown. In this case, a statue has been duplicated. The top row shows all the keypoints, while the bottom row shows only those matching according to the CMFD. Again, in the leftmost column, we can observe that the CMFD detects the forgery, but after keypoint removal, the keypoints in the area are drastically reduced, thus lowering the number of detected matches to 2 (versus a threshold of 4) in a way that leads the CMFD to

Table 2 Keypoints and SIFT matches when copy-move is carried out with/without keypoint removal and with keypoint removal-injection

	No keypoint removal	Keypoint removal	Keypoint removal-injection
Church			
Keypoints	10	0	3
Matches	8	0	0
Nativity			
Keypoints	62	5	14
Matches	41	2	3

Figure 9 Example of copy-move removal-injection attack (2). Extracted keypoints (top) and SIFT matches (bottom) for image *Nativity*: copy-move attack without keypoint removal (left), with keypoint removal (middle), and with keypoint removal-injection (right).

a failure. The result following keypoint injection is displayed the rightmost column of Figure 9. Keypoints are actually reinserted (see Table 2 for numerical details), but SIFT matches only increase to 3 (hence, still under CMFD threshold), and no forgery is detected by the tool.

In conclusion, our examples demonstrate that it is indeed possible to apply the keypoint removal-injection procedure to produce an image that is forensically undetectable by the targeted CFMD (thanks to removal) and that does not exhibit traces of counter-forensic processing (thanks to injection).

6 Conclusions

In this paper, we considered the injection of fake SIFT keypoints on an image whose authentic keypoints have been previously deleted. So far, no systematic study on this topic was carried out. Our interest stemmed from the consideration that an image with *too few* keypoints is per se a clue of counterfeit, which can be used to reveal the removal attack. As a consequence, the adversary must improve the imperceptibility of the removal attack by introducing new keypoints that are detectable by SIFT but do not have any correspondence with the authentic ones. With this aim, we presented a procedure to firstly remove and then reinsert keypoints by resorting to a set of possible injection tools. Experimental results on a large data set are encouraging and already show that injection is feasible without causing a successive detection at SIFT-match level. Visual quality is still preserved both with respect to the original image and, particularly, in comparison with the image quality achieved after keypoint removal only. Moreover, we applied the proposed removal-injection procedure to impair a state-of-the-art SIFT-based copy-move detector; we removed the keypoints matching across cloned areas to prevent detection and successively repopulated the attacked region with fake keypoints to cover the manipulation. Future works will be dedicated to (1) the research of more effective and *ad hoc* injection tools allowing, for example, the choice of the spatial location and scale by which the fake keypoints are injected and (2) a detailed evaluation of the perceptibility of the removal-injection from the perspective of the forensic analyst. Currently, in fact, no other solution than visual inspection exists to expose adversarial processing on SIFT keypoints.

Competing interests
The authors declare that they have no competing interests.

Authors' information
IA received her *Laurea* degree in Computer Engineering in 2006 and her 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 post-doctoral 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 his 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 to about 270 papers and holds four patents in the field of digital watermarking and document protection. He is the 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 his *Laurea* degree in Electronic Engineering in 1997 and his 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 Sciences 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 and by the SECURE! Project, funded by the POR CreO FESR 2007-2013 programme of the Tuscany Region (Italy).

Author details
[1]Department of Information Engineering and Mathematical Sciences, University of Siena, Via Roma, 56, Siena 53100, Italy. [2]Media Integration and Communication Center, University of Florence, Viale Morgagni, 65, Florence 50134, Italy.

Received: 11 October 2013 Accepted: 5 December 2013
Published: 20 December 2013

References

1. V Christlein, C Riess, J Jordan, C Riess, E Angelopoulou, An evaluation of popular copy-move forgery detection approaches. IEEE Trans. Inf. Forensics Secur. **7**(6), 1841–1854 (2012)
2. X Pan, S Lyu, Region duplication detection using image feature matching. IEEE Trans. Inf. Forensics Secur. **5**(4), 857–867 (2010)
3. I Amerini, L Ballan, R Caldelli, A Del Bimbo, G Serra, A SIFT-based forensic method for copy move attack detection and transformation recovery. IEEE Trans. Inf. Forensics Secur. **6**(3), 1099–1110 (2011)
4. DG Lowe, Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
5. R Böhme, M Kirchner, Counter-forensics: attacking image forensics, in *Digital Image Forensics (Chapter 10)*, ed. by HT Sencar, N Memon (Springer, New York, 2012)
6. RC Veltkamp, M Tanase, Content-based image retrieval systems: a survey. Technical report, Utrecht University (2002)
7. CY Hsu, CS Lu, SC Pei, Secure and robust SIFT, in *Proceedings of the 17th ACM International Conference on Multimedia, MM '09* (ACM, New York, 2009), pp. 637–640
8. TT Do, E Kijak, T Furon, L Amsaleg, Understanding the security and robustness of SIFT, in *Proceedings of the 18th ACM International Conference on Multimedia, MM '10* (ACM, New York, 2010), pp. 1195–1198
9. TT Do, E Kijak, T Furon, L Amsaleg, Deluding image recognition in SIFT-based CBIR systems, in *Proceedings of the 2nd ACM Workshop on Multimedia in Forensics, Security and Intelligence, MiFor '10* (ACM, New York, 2010), pp. 7–12
10. TT Do, E Kijak, L Amsaleg, T Furon, Enlarging hacker's toolbox: deluding image recognition by attacking keypoint orientations, in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP12)* (IEEE, Piscataway, 2012), pp. 1817–1820
11. R Caldelli, I Amerini, L Ballan, G Serra, M Barni, A Costanzo, On the effectiveness of local warping against SIFT-based copy-move detection, in *Proceedings of the International Symposium on Communications, Control and Signal Processing (ISCCSP12)* (IEEE, Piscataway, 2012), pp. 1–5
12. A D'Angelo, M Barni, A structural method for quality evaluation of desynchronization attacks in image watermarking, in *Proceedings of the IEEE 10th Workshop on Multimedia Signal Processing* (IEEE, Piscataway, 2008), pp. 754–759
13. I Amerini, M Barni, R Caldelli, A Costanzo, Counter-forensics of SIFT-based copy-move detection by means of keypoint classification. EURASIP J. Image Video Process. **2013**, 18 (2013)
14. DG Lowe, Object recognition from local scale-invariant features, in *Proceedings of the 7th IEEE International Conference on Computer Vision,* Kerkyra, September 1999, vol. 2 (IEEE, Piscataway, 1999), pp. 1150–1157
15. T Hastie, R Tibshirani, JJH Friedman, *The Elements of Statistical Learning*, vol. 2 (Springer, New York, 2001)
16. MJ Swain, DH Ballard, Color indexing. Int. J. Comput. Vis. **7**, 11–32 (1991)
17. K Zuiderveld, Contrast limited adaptive histogram equalization, in *Graphics Gems IV* (Academic Press Professional, Inc., San Diego, 1994), pp. 474–485. [http://dl.acm.org/citation.cfm?id=180895.180940]
18. D Sheet, H Garud, A Suveer, M Mahadevappa, J Chatterjee, Brightness preserving dynamic fuzzy histogram equalization. IEEE Trans. Consum. Electron. **56**(4), 2475–2480 (2010)
19. P Perona, J Malik, Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **12**(7), 629–639 (1990)
20. J Weickert, Coherence-enhancing diffusion filtering. Int. J. Comput. Vis. **31**(2), 111–127 (1999)
21. J Weickert, H Scharr, A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance. J. Vis. Commun. Image Representation **13**, 103–118 (2002)
22. Z Wang, AC Bovik, HR Sheikh, EP Simoncelli, Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
23. A Vedaldi, B Fulkerson, VLFeat: an open and portable library of computer vision algorithms, in *Proceedings of the ACM International Conference on Multimedia* (ACM, New York, 2010), pp. 1469–1472
24. G Schaefer, M Stich, UCID: an uncompressed color image database, in *Storage and Retrieval Methods and Applications for Multimedia 2004*, vol. 5307 (International Society for Optics and Photonics, Bellingham, 2003), pp. 472–480

doi:10.1186/1687-417X-2013-8
Cite this article as: Amerini *et al.*: Removal and injection of keypoints for SIFT-based copy-move counter-forensics. *EURASIP Journal on Information Security* 2013 **2013**:8.