Forensic Analysis of SIFT Keypoint Removal and Injection

Andrea Costanzo, Irene Amerini, Roberto Caldelli, *Member, IEEE*, and Mauro Barni, *Fellow, IEEE*

***Abstract*— Attacks capable of removing SIFT keypoints from images have been recently devised with the intention of compromising the correct functioning of SIFT-based copy–move forgery detection. To tackle with these attacks, we propose three novel forensic detectors for the identification of images whose SIFT keypoints have been globally or locally removed. The detectors look for inconsistencies like the absence or anomalous distribution of keypoints within textured image regions. We first validate the methods on state-of-the-art keypoint removal techniques, then we further assess their robustness by devising a counter-forensic attack injecting fake SIFT keypoints in the attempt to cover the traces of removal. We apply the detectors to a practical image forensic scenario of SIFT-based copy-move forgery detection, assuming the presence of a counterfeiter who resorts to keypoint removal and injection to create copy–move forgeries that successfully elude SIFT-based detectors but are in turn exposed by the newly proposed tools.**

***Index Terms*— Image forensics, counter-forensics, SIFT, keypoint removal, keypoint injection, copy-move forgery detection.**

I. Introduction

OVER the last few years, increasing attention has been devoted to counter-forensics, that is the discipline aiming at hindering forensic analysis by taking advantage of its limits. Despite the relative youth, counter-forensic literature already offers a number of techniques to conceal relevant footprints like the artefacts introduced by lossy compression, resampling or histogram manipulations [1]; by relying on such techniques, a clever counterfeiter can create forgeries that are undetectable by the targeted forensic algorithms.

Copy-move forgery, whereby a portion of the image is cloned once or more times to either hide or introduce semantically relevant content, is one of the most common ways to alter the message conveyed by an image [2]. Among the most robust copy-move detectors there are those based on the Scale Invariant Feature Transform (SIFT) [3], whose capability to discover correspondences between similar content allows a fast and accurate detection of cloned areas [4], [5].

Although resilient to several processing, state-of-the-art techniques can be successfully challenged by manipulating SIFT features (or keypoints) so to prevent the match between cloned image regions. In general, such objective is pursued either by altering the keypoint neighbourhoods in such a way that keypoint descriptors of corresponding points in cloned areas do not match anymore or by directly removing the keypoints from the cloned regions; to the best of our knowledge, the attacks to SIFT-based copy-move detection proposed so far belong to the latter category [6], [7] and although there are no doubts about their effectiveness, the statistical detectability of this kind of attacks should concern the attacker. Like any other processing, in fact, keypoint removal leaves traces into the manipulated areas, in the form of high-variance textured regions where keypoints should be found but are instead absent. Despite that, no method has been devised so far to find such footprints. To take advantage of these footprints and restore the efficiency of SIFT-based forensic analysis, it is recommendable to devise adversary-aware algorithms to understand whether SIFT keypoints have been artificially removed.

A. Previous Works

To the best of our knowledge, the first study on SIFT security is the one by Hsu *et al.* [8], where an authentication system based on SIFT and image hashing was bypassed by deleting keypoints. The technique in [8] was later applied by Do *et al.* against Content Based Image Retrieval [9] and, despite being ineffective in such a scenario, it inspired new attacks to SIFT spatial locations [10] and dominant orientations [11]. The purpose of these attacks was to cause a wrong answer to an image query by removing or altering the keypoints of the queried image so to decrease its similarity score with the data base entries. A similar approach was adopted in [12] by Lu *et al.* to disable an image copy detection system. The elimination of large amounts of keypoints has been investigated in-depth also in the context of image forensics. A first attempt in this sense is described in [6], where warping attacks typical of watermarking were used to bypass SIFT-based copy-move detection. A more powerful attack against the same category of detectors has been proposed in [7], where existing and new algorithms are combined to

Manuscript received January 7, 2014; revised May 10, 2014; accepted July 7, 2014. Date of publication July 10, 2014; date of current version July 31, 2014. This work was supported in part by the Rewind Project through the Future and Emerging Technologies Programme within the 7FP of the European Commission under Grant 268478 and in part by the Secure Project through the POR CreO FESR 2007-2013 Programme, Tuscany, Italy. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Chiou-Ting Hsu.

A. Costanzo and M. Barni are with the Department of Information Engineering and Mathematics, University of Siena, Siena 53100, Italy, and also with the National Inter-University Consortium for Telecommunications, Parma 43124, Italy (e-mail: andreacos82@gmail.com; barni@dii.unisi.it).

I. Amerini is with the Media Integration and Communication Center, University of Florence, Florence, 50134, Italy (e-mail: irene.amerini@unifi.it).

R. Caldelli is with the National Inter-University Consortium for Telecommunications, Parma 43124, Italy (e-mail: roberto.caldelli@unifi.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2014.2337654

remove an arbitrary amount of keypoints by preserving at the same time the overall image quality. Recently, the same authors approached the dual problem of re-introducing forged keypoints by using adaptive image enhancing algorithms as a means to conceal the traces of keypoint removal [13].

B. Contributions

The main contribution of this paper consists in three forensic detectors for the identification of images whose SIFT keypoints have been artificially removed and, possibly, re-inserted. The proposed algorithms scan image regions with sufficiently high variance in search of suspect inconsistencies in the number and in the distribution of SIFT keypoints. By relying on such algorithms, the forensic analyst can decide on the authenticity of the image as a whole or localise tampered regions within the image by means of a sliding window approach.

We studied the performance of the detectors in two different cases of increasing complexity, when only keypoint removal is carried out and when fake keypoints are re-introduced into the image to hide the traces of the preceding removal. For the latter case, we also developed a new scheme for keypoint injection, which represents the second contribution of this work. Such an attack is based on the classification of image regions (salient/non-salient) followed by a set of injection algorithms specifically tailored to each class. The attack outperforms the injection techniques discussed in [13], thus representing a harder challenge for the proposed detectors. The results we obtained show that, regardless of the attack, we can effectively discriminate between authentic and forged images.

We demonstrate the usefulness of the keypoint removal detectors in a SIFT-based copy-move detection scenario. In particular, we show how a counterfeiter who resorts to keypoint removal and injection to successfully bypass copy-move detection [5] is unable to evade the exposure of the forgery by means of the proposed algorithms.

The outline of this paper is the following. Sec. II briefly summarises the working principles of the state-of-the-art attack that we will be using to remove SIFT keypoints; Sec. III describes the detectors and Sec. IV experimentally validates them; Sec. V introduces a new algorithm to inject fake SIFT keypoints; Sec. VI evaluates the injection capability of the proposed scheme; finally, in Sec. VII all the above techniques are evaluated in the context of copy-move forgery detection.

II. SIFT Keypoint Removal

Throughout the paper we will assume that the reader is familiar with the theoretical and technical aspects of the SIFT algorithm, for which we refer to [3]. Here we briefly outline the working principles of the SIFT keypoint removal scheme, called Classification-Based Attack (CLBA) [7], which we are going to use for the rest of the paper.

A. Classification-Based Keypoint Removal

The rationale behind CLBA is that not all SIFT keypoints have the same properties and thus their robustness to certain manipulations aiming at removing them varies. Therefore, CLBA resorts to keypoint classification preceding the manipulation itself, in such a way that each class is addressed by one or more attacks specifically tailored to exploit its weaknesses.

More specifically, keypoints are classified according to the number of modes of the grayscale histogram of the neighbourhood surrounding them in the pixel domain. A keypoint can be assigned to one among three classes, i.e. unimodal, bimodal and multimodal, each corresponding to a different class of visual contents: respectively, uniform low-variance regions, edges or geometric shapes and noisy high-variance regions. Depending on the class, each keypoint is removed by manipulating a small square region (referred to as support) around it by means of one or more of the following attacks: Gaussian Smoothing (GS), Collage and Removal with Minimum Distortion (RMD) [10]. GS flattens the local pixel intensities in such a way that the contrast value of keypoints slightly above the SIFT threshold is artificially decreased. The Collage attack consists in the substitution of the keypoint neighbourhood with a new neighbourhood not containing a keypoint, chosen from a database according to a similarity criterion. The RMD attack relies on a constrained optimisation problem to determine coefficients that are added to the keypoint neighbourhood to artificially decrease the local contrast value below SIFT acceptance threshold.

CLBA arranges the above attacks into an iterative procedure. At the first iteration, a grayscale image is fed to the system, which starts by detecting SIFT keypoints. Then, the neighbourhood of each keypoint is assigned to a class and attacked accordingly. Once all first-iteration keypoints have been attacked, the procedure is iterated on the manipulated image. Intuitively, earlier iterations are aimed to remove weaker keypoints by means of GS, whereas later iterations deal with robust keypoints by means of the more effective Collage and RMD. The iterations continue until a target condition on the percentage of removed keypoints, the minimum allowed image quality or the maximum number of iterations, is satisfied. The advantages of such an iterative approach reside in the fact that it allows to adjust the strength of the attacks as more robust keypoints keep surviving and to deal with the keypoints that may be introduced into the image as a side effect of removal. In CLBA, in fact, new keypoints accidentally generated during an iteration are classified and attacked again at the subsequent iterations. For a detailed description of the above concepts, we refer to [7].

Note that CLBA addresses only the first-octave keypoints, i.e. those extracted from the image at its original resolution. This fact prevents CLBA from being a threat to applications relying on the fewer but more robust higher-octave keypoints. The attack, however, can effectively counter copy-move detection, whereby the majority of matches linking the cloned regions are extracted from the first octave.

B. Performance of Keypoint Removal

CLBA's effectiveness is evaluated in terms of keypoint removal rate (KRR), i.e. the percentage of keypoints that are successfully deleted, whereas its perceptibility is evaluated in terms of average Peak Signal-to-Noise Ratio (PSNR) and Structural SIMilarity (SSIM) index [14] between the manipulated and the original neighbourhoods. In [7], it was shown that CLBA outperforms the other class-unaware removal attacks proposed thus far (see [9], [10]) and ensures a good trade-off between effectiveness and imperceptibility.

In this work we make use of a more powerful variant of the original algorithm based on two supports of increasing size, whereby the image is first attacked with support 8 × 8 to remove the majority of keypoints and then attacked again with support 10 × 10 to remove those keypoints that were not affected by the first manipulation. The first stage of the attack is halted after 25 iterations, the second stage after 40. This modification yields higher removal rates than the original version (99.1% versus 92% average KRR on the same image data set used in [7]) without further deteriorating the visual quality of the forgery.

When practical applications require a larger amount of keypoints, SIFT can upscale the image by a factor 2 before extraction [3]. These keypoints, however, are not robust against CLBA and thus can be removed (99.8% KRR) with less iterations (15) and smaller attack supports (4 × 4 pixels). In this case, one can first remove all the keypoints of the octave -1, then iterate again CLBA on the keypoints of the octave 0. For the sake of simplicity, in this paper we will consider only the first-octave keypoints.

Before moving to the next section, it is worth mentioning that the keypoint removal detectors we are going to present do not depend on a specific removal algorithm, whose choice is thus arbitrary. Other methods than the one we adopted could be employed such as, for example, the one in [12]. In [12], the authors cast keypoint removal as a constrained optimisation problem minimising image distortion and obtain KRR values comparable to those in [7].

III. Detection of SIFT Keypoint Removal

A. Keypoint-to-Corner Ratio Detector

Our first keypoint removal detector is based on two simple observations. The first observation is that SIFT keypoints lie in proximity of corners, i.e. interest points where two edges intersect, which can be identified by means of methods like the Harris [15] or the Shi and Tomasi [16] detectors. Such an observation is supported both by SIFT theory, which selects keypoints according to a corner response metric inspired to the Harris detector, and by the experimental results that we will discuss in Sec. IV-C.

The second observation is that all removal attacks devised so far have been designed to preserve as much as possible image content by working on small neighbourhoods of the keypoints. As a consequence, while the number of keypoints is significantly reduced by a removal attack, the number and the pixel-domain locations of corners are subject to negligible variations. In Sec. IV-C we demonstrate experimentally the validity of these assumptions. We believe that a possible theoretical explanation of this fact is the following. At the detection stage, a DoG extremum is considered a stable keypoint if both the local contrast and the corner response are higher than a threshold. Two of the attacks composing CLBA (i.e. RMD and Gaussian Smoothing) are designed to lower the local contrast so that the keypoint is rejected based on the first check and consequently they tend to leave the corner response unaltered.

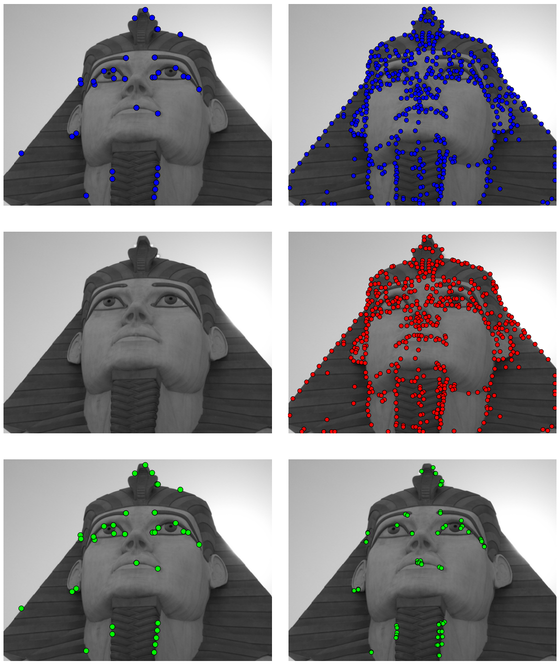


Fig. 1. Impact of keypoint removal on keypoints (left column) and corners (right column) distribution. First row: before CLBA; second row: after CLBA; third row: difference in corners and keypoints.

If the above two assumptions hold, then we can understand whether an image has been subject to a keypoint removal attack based on the conservation of corners and on the reduction of keypoints in proximity of corners. Let $N_{corners}$ be the number of corners; consider a square patch of side d centred on each corner and let $N_{keypoints}$ be the total amount of keypoints falling into all such patches. An image is labelled as forged if the ratio between the above two quantities (Keypoint-to-Corner Ratio or KCR) falls below a threshold:

$$KCR = \log_{10}\left(\frac{N_{keypoints}}{N_{corners}}\right) \overset{?}{\leq} T_1. \quad (1)$$

Under the hypothesis that keypoint removal does not affect corners, the denominator of (1) is approximately the same for the authentic and the manipulated image. Conversely, the numerator of (1) is drastically reduced by the attack, hence the KCR index of the attacked image should be smaller than that of its authentic counterpart. The value d defining the corner proximity as well as the threshold T_1 are empirically derived in Secs. IV-C and IV-E.

Even though technical details are left to the sequel, in Fig. 1 we show an example of the population of SIFT keypoints (left column) and Harris corners (right column) before and after CLBA (respectively, first and second row), in order to emphasise the conservation of corners and the reduction of keypoints. To help understanding the consequences of the attack, we highlighted the differences in the last row of Fig. 1,

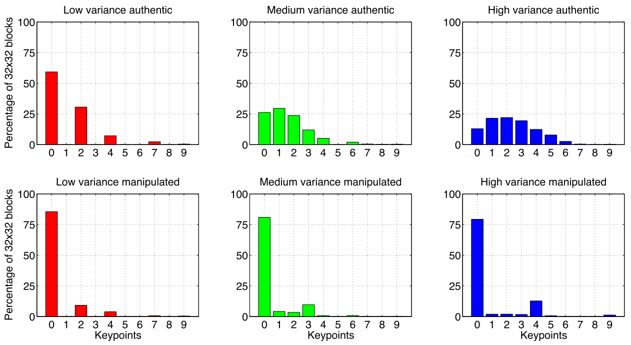


Fig. 2. Histograms h_L, h_M and h_H for a test image. Top row: authentic; bottom row: forged.

where the green markers identify the deleted keypoints (left) and corners (right). The KCR index is respectively -1.6 for the authentic image and -15.7 for the forgery.

B. CHI-Square Distance Detector

The second detector is based on the observation that keypoints are concentrated in image regions characterised by high variance, due to the fact that SIFT discards candidate keypoints whose neighbourhoods have low contrast. Consequently, an image targeted by keypoint removal should exhibit high variance regions unnaturally deprived of keypoints. To translate this intuition into a detectable footprint, we studied the distributions of SIFT keypoints in image blocks characterised by different variance. In practice, first we assigned each non overlapping block of side B of the image to one among three classes depending on its variance: *low*, *medium* and *high* (see Sec. IV-A for technical details on variance-based classification); then, we examined each class in search of anomalies in the distribution of keypoints. More specifically, we adopted the following strategy.

1) Divide an image I into non-overlapping 32×32 blocks whose variance is assigned either to the *low*, *medium* or *high* class.
2) For each class of variance and given a fixed amount of bins (10 in our implementation), compute the percentage of blocks containing a certain number of keypoints.[1] The resulting percentages correspond to histograms referred to as h_L, h_M and h_H.
3) Attack the image with CLBA and repeat steps 1)–2) on the forged image.

The result of the above procedure is shown in Fig. 2, where the first row corresponds to the histograms of an authentic image and the second row to those of its tampered version. As a matter of fact, we can recognise different shapes, especially in the case of medium and high variance histograms, where the attack has generated an anomalous percentage of textured blocks without keypoints.

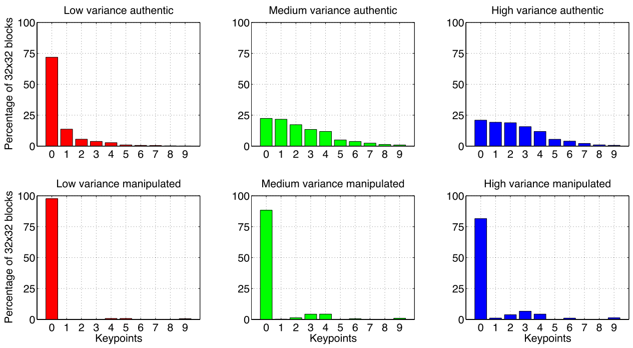


Fig. 3. Accumulated reference histograms for the CHI detector. Top row: authentic; bottom row: forged.

Since one image alone is not sufficient to draw any conclusion on the reliability of the above footprint, we repeated the same procedure on a large data set of images (see Sec. IV-A). We accumulated each class histogram of all the images with a bin-by-bin sum and we averaged them, first for the authentic images and then for their forged versions, as follows:

$$\widehat{H}_{\mathcal{C}_k}^{(auth)} = \frac{1}{N_{auth}} \sum_{i=1}^{N_{auth}} h_{\mathcal{C}_k}^{(i)} \tag{2}$$

$$\widehat{H}_{\mathcal{C}_k}^{(forged)} = \frac{1}{N_{forged}} \sum_{j=1}^{N_{forged}} h_{\mathcal{C}_k}^{(j)}, \tag{3}$$

where: $h_{\mathcal{C}_k}^{(i)}$ is the histogram of blocks belonging to the variance class $\mathcal{C}_k$ for the i-th image; N_{auth} and N_{forged} are the number of authentic and forged images, respectively.

The resulting histograms are shown in Fig. 3; the trends we observed on a single image are now even more evident. We will refer to the accumulated histograms as $\widehat{H}_L^{(auth)}$, $\widehat{H}_M^{(auth)}$ and $\widehat{H}_H^{(auth)}$ for authentic images and as $\widehat{H}_L^{(forged)}$, $\widehat{H}_M^{(forged)}$ and $\widehat{H}_H^{(forged)}$ for forged images. We will consider such histograms our *ground truth*.

The shape difference for *low* variance histograms is not as pronounced as for the other classes because candidate keypoints in uniform regions are less likely to pass the SIFT contrast check. Therefore, we rely only on the other two histograms. In particular, we look at the distance between the *medium* variance histogram of the image under analysis and $\widehat{H}_M^{(auth)}$ and compare it against a threshold. In our implementation we chose the chi-square distance [17], hence the name CHI detector:

$$\chi^2 = \frac{1}{2} \sum_{l=1}^{L} \frac{\left(h_M(l) - \widehat{H}_M^{(auth)}(l)\right)^2}{h_M(l) + \widehat{H}_M^{(auth)}(l)} \overset{?}{\geq} T_2, \tag{4}$$

where l denotes one histogram bin and L indicates the number of bins. The image is considered as forged if the distance exceeds T_2, whose value is empirically derived in Sec. IV-E.

Note that in (4) we chose the histogram of *medium* variance blocks but, according to our experiments, similar performance can be attained by considering the histogram of *high* variance blocks. Nevertheless, we prefer the former because it is more

[1] For example, in the case of the *medium* class, for $bin = 0$, we count the percentage of medium blocks containing 0 keypoints; for $bin = 1$, the percentage of medium blocks containing 1 keypoint and so on, until we reach the last bin.

reliable on images characterised by few textured regions and consequently by few blocks with high variance.

C. SVM Detector

The last detector is based on a Support Vector Machine (SVM) learning model discriminating between authentic and CLBA-forged images. The feature vector $\mathbf{F}$ that we used to represent the examples of each class is obtained by concatenating the bin values of the histograms h_L, h_M and h_H and thus consists of 30 elements. We used a large amount of examples $\mathbf{F}$ coming from authentic and forged images to train a probabilistic SVM [18]. By resorting to such model, in fact, we can determine not only the class to which a new example (i.e. the image under analysis) belongs, but also its probability of belonging to the assigned class. Consequently, the analysed image I is labelled as tampered if the SVM output O_{svm} is higher than a threshold:

$$O_{svm} = Prob(I \text{ is forged}) \overset{?}{\geq} T_3. \tag{5}$$

In Sec. IV-D we provide all the technical details of the adopted SVM and in Sec. IV-E we experimentally determine the value of T_3. Once we introduced the notion of injection of fake SIFT keypoints, in Sec VI-C we extend the SVM to a multi-class model in such a way to discriminate between authentic, CLBA-forged and injected images.

IV. Experimental Validation of Keypoint Removal Detectors

A. Experimental Setup

Variance-based classification is carried out on square blocks of side $B = 32$. The value of each element $V(i, j)$ is the variance within a square window centred on the corresponding image pixel $I(i, j)$. We first binarise V and then we subdivide both V and I into non overlapping $B \times B$ blocks. Let I_B be one of such image blocks and V_B the corresponding variance block; the ratio τ of pixels having value 1 with respect to the total number of pixels of V_B is computed and I_B is classified as follows: if $0 \leq \tau \leq \frac{1}{3}$, then I_B is *low*; if $\frac{1}{3} < \tau \leq \frac{2}{3}$, then I_B is *medium*; and if $\frac{2}{3} < \tau \leq 1$, then I_B is *high*.

First-octave keypoints are removed by means of CLBA with 100% target removal rate and a maximum of 40 iterations, unless specified otherwise. For the detection of SIFT features we rely on the popular VLFeat libraries [19], with edge and peak-difference thresholds set to 10 and 4 respectively.

It is important to point out that the results we show are obtained by taking into account only the keypoints belonging to first octave (i.e. octave 0), i.e. those deleted by the CLBA. The conclusions drawn for the detectors, however, are general and retain their validity on all octaves.

B. Image Data Sets

We collected two image data sets to test the algorithms, one consisting of the first 1000 images of the well-known INRIA Holidays data set [20] and one consisting of the first 100 images of the INRIA Copydays data set. In the sequel, we will refer to the data sets as *Holidays1000* and *Copydays100*.

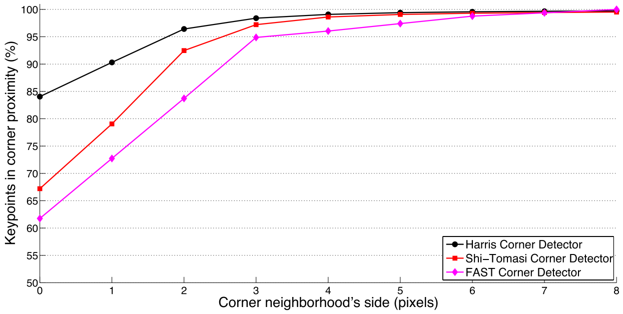


Fig. 4. Percentage of keypoints in the $d \times d$ neighbourhood of corners for three different corner detectors.

To limit the complexity of the experiments, we downscaled all the images to 1600 × 1200 pixels. We have chosen fairly large images because the amount of keypoints (on average, in the order of 2500 for both data sets) allows us to assess more accurately the performance of the detectors; however, in Sec. IV-E we also consider smaller images.

The idea behind the two data sets is to use the *Holidays1000* for the verification of the hypotheses underlying the detectors, for parameter tuning (e.g. determining thresholds) and to train the SVM. Once we gathered such information, the three algorithms are tested on the *Copydays100*.

C. Verification of the Hypotheses Behind the Detectors

The proposed detectors rely on hypotheses that require the support of experimental data. In particular, we need to verify that SIFT keypoints lie in proximity of corners and that keypoint removal does not affect significantly the number and the position of corners nor variance-based block classification.

1) SIFT Keypoints Lie in Proximity of Corners: For all the images of the *Holidays1000* data set, we measured the percentage of SIFT keypoints falling within a $d \times d$ neighbourhood of the corners provided by three detectors: Harris [15], Shi and Tomasi [16] and FAST (Features from Accelerated Segment Test) [21]. We set the free parameter k, the response threshold R and the width of the derivative filter σ of the Harris detector to 0.04, 1000 and 3 respectively; the same R and σ were used for the Shi and Tomasi detector; for the FAST detector, we considered circular regions of 16 pixels centred on the candidate corners and corner threshold 20.

When $d = 0$, we count only the number of keypoints coinciding with corners; as d increases, we also include the keypoints in the proximity of the corner. Clearly, a keypoint is counted only once even if it falls in the neighbourhood of more than one corner. Fig. 4 shows the result of this experiment averaged over all the images. The percentage of keypoints coinciding with corners is lower for the Shi and Tomasi and the FAST detectors because they provide less numerous but more robust corners. However, regardless of the algorithm, on average more than 95% of the keypoints of an image are contained in small neighbourhoods of side $d = 3$, thus confirming our starting assumption.

2) Keypoint Removal Does Not Affect Corners: Fig. 5 reports the difference in the number of Harris corners before

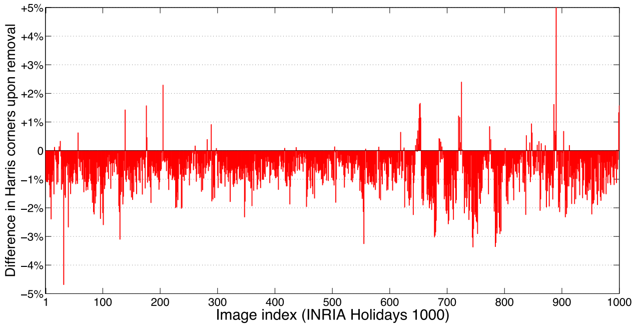


Fig. 5. Difference in the number of Harris corners following CLBA.

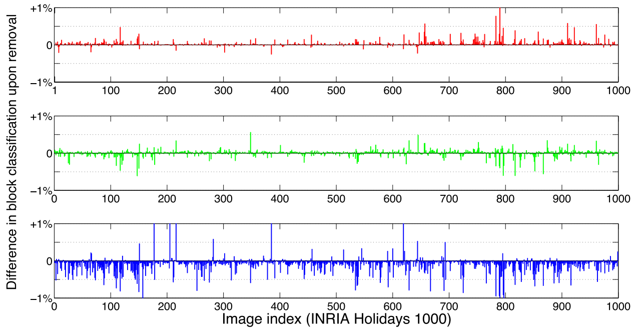


Fig. 6. Difference in block-based variance classification following CLBA. Top: *low* variance; middle: *medium* variance; and bottom: *high* variance.

and after CLBA on the *Holidays1000* data set (given the similarity, we omit the other corner detectors for sake of brevity). For each image, we express the difference between the number of corners of the authentic and CLBA-forged images in terms of percentage. The reduction in the number of corners in the majority of the images is due to the smoothing operators used by the removal attack; nevertheless, for 97% of the images the difference is bounded in $[+1\%, -3\%]$.

Furthermore, to verify whether keypoint removal modifies the spatial location of corners, for each image of the data set we proceeded as follows. First, we computed the Euclidean distance between each original corner and the corners in the CLBA-forged image, extracted by means of the Harris detector. Then, we evaluated the distance d_c of the closest corner: if $d_c = 0$, then the corner under analysis is not affected by CLBA. The results we obtained by averaging over all corners and images support our assumption on the robustness of corners, as 85% of corners conserve their spatial location. It is also worth mentioning that the new location of the altered corners is very close to the original one (average Euclidean distance 1.44 pixels).

3) Keypoint Removal Does Not Affect Block Classification: The procedure leading to the graphs of Fig. 6 is the same as above. This time, however, we plot the differences in number of *low*, *medium* and *high* variance blocks before and after the removal attack. Again, variations are not significant, being confined to the interval $[-1\%, +1\%]$ for all classes.

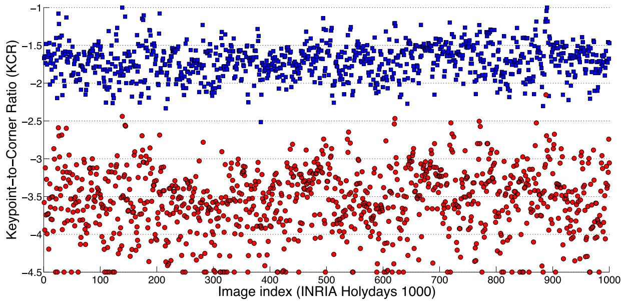


Fig. 7. KCR scattergram. Blue squares: authentic images red circles: CLBA-forged images.

D. Detection of Full-Frame Keypoint Removal

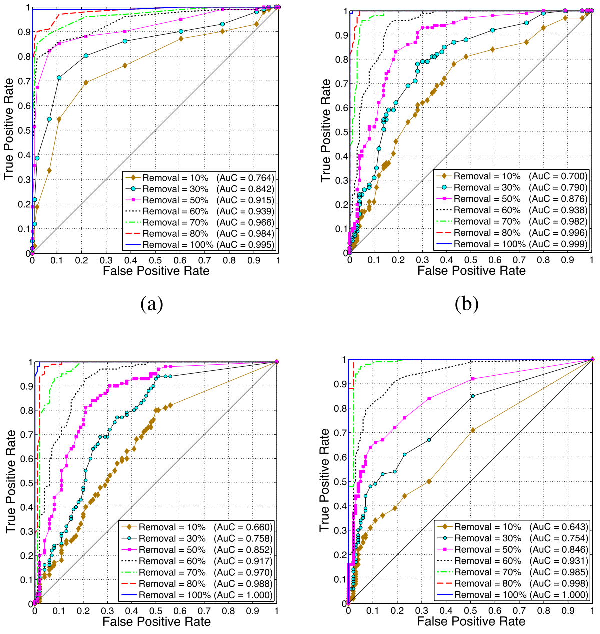
The contribution of this section is twofold: first, we analyse and compare the performance of each detector by means of Receiving Operator Characteristics (ROC); then, we derive the thresholds ensuring an acceptable trade-off between the probabilities of detection and false alarm.

1) KCR Detector: We extracted corners by means of Shi and Tomasi's algorithm [16] for two reasons: 1) its corners are more robust and less numerous than those extracted by the Harris detector, thus providing higher KCR scores for authentic images[2]; and 2) the significantly lower time complexity with respect to the FAST detector. The performance of KCR based on other detectors, however, are only marginally inferior to those granted by the chosen corner extractor. In accordance with the data of Sec. IV-C, we set the corner neighbourhood side to $d = 3$.

To assess the discriminative power of the KCR index, we computed the scattergram shown in Fig. 7. In practice, the index is calculated for all the images of the *Holidays1000* data set and their forged versions (respectively, blue squares and red circles). The observations sitting exactly on the x-axis correspond to the images with very low KCR, which we set to -4.5 for better readability. It comes out that scores cluster into two distinct and separable groups, as confirmed by the ROCs of Fig. 8 (a). We obtained each curve by varying the threshold separating the clusters in the interval $[0, -5]$ (step 0.01) and CLBA's target removal rate in the interval $[10\%, 100\%]$ (step 10), in order to understand to what extent the forgery is detectable. Expectedly, the more keypoints are left into the image, the harder the separation of the clusters; regardless of that, acceptable results can be obtained even for very low removal rates.

2) CHI Detector: We validated the CHI detector by using the same procedure of KCR. Because the behaviour is very similar, we do not show the scattergram for the χ^2 distance. The curves of Fig. 8 (b) and (c) have been calculated on the *Copydays100* data set by relying on the reference histograms obtained from the *Holidays1000* (Sec. III-B, Fig. 3) and by varying the threshold T_2 in $[0, 80]$ (step 1) and the target removal rate in $[10\%, 100\%]$. We show two ROCs: the one on the top right is based on the χ^2 distance from $\widehat{H}_M^{(auth)}$

[2] The higher amount of Harris corners, in fact, tends to unnecessarily increase the denominator of (1), thus making the discrimination between authentic and CLBA-forged images harder.



(a) (b)

(c) (d)

Fig. 8. ROCs depending on removal rate: (a) KCR; (b) CHI, χ^2 distance from $\widehat{H}_M^{(auth)}$; (c) CHI, χ^2 distance from $\widehat{H}_M^{(forged)}$; and (d) SVM.

TABLE I

AUC FOR THE DETECTORS AS A FUNCTION OF REMOVAL RATE

Detector	100%	80%	70%	60%	50%	30%	10%
KCR	0.99	0.98	0.96	0.93	0.91	0.84	0.76
CHI	0.99	0.99	0.98	0.93	0.87	0.79	0.7
SVM	1	0.99	0.98	0.93	0.85	0.75	0.63

and the one on the bottom left on the χ^2 distance from the manipulated medium reference $\widehat{H}_M^{(forged)}$; both solutions are equally viable.

3) SVM Detector: The SVM detector is implemented by relying on the well-established LIBSVM software [22]. The training stage has been carried out on 2000 feature vectors **F**, i.e. those extracted from the *Holidays1000* data set and its CLBA-forged version. Recall that each feature vector consists of 30 elements, i.e. the 10 bins for *low*, *medium* and *high* variance histograms. The tests have been carried out on the 200 feature vectors coming from the *Copydays* data set. We used a Radial Basis Function kernel with parameters $C = 8$ and $\gamma = 0.5$ derived from a 5-fold cross-validation on 400 images, which were not considered again for training to prevent over-fitting. The SVM model is probabilistic, i.e. it outputs the probability that the image under analysis is tampered. We calculated each ROC curve of Fig. 8 (d) by varying T_3 in [0, 1] (step 0.01) and the target removal rate as in the previous tests.

To conclude this set of experiments, in Tab. I and Tab. II we compare the detectors in terms of Area Under Curve (AUC) and true positive rate (fixed false positive rate of 0.15), as a function of keypoint removal rate; noticeable differences in performance exist only for removal rates below 50%, for which the KCR detector appears to be the most reliable.

TABLE II

TRUE POSITIVE RATE FOR THE DETECTORS FOR A FALSE POSITIVE RATE 0.15 AS A FUNCTION OF REMOVAL RATE

Detector	100%	80%	70%	60%	50%	30%	10%
KCR	0.99	0.98	0.94	0.87	0.86	0.75	0.6
CHI	0.99	0.99	0.97	0.75	0.52	0.28	0.21
SVM	1	1	0.99	0.86	0.68	0.53	0.37

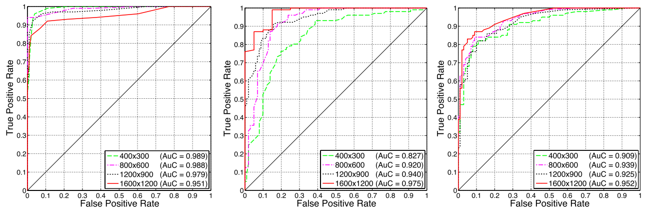


Fig. 9. ROC for mixed removal rates depending on image size. From left to right: KCR, CHI and SVM.

E. Dependence on Image Size and Removal Rate

While assigning the values of the thresholds T_1, T_2 and T_3, we took into account two factors: the target removal rate and the image size, as both parameters may impact on the performance of the detectors: low removal rates cause the forged image to be similar to the authentic one; image resizing reduces the number of keypoints and corners and thus it could alter their ratio.

In the following experiment we used 100 images randomly drawn from the *Holidays1000* to build 4 new data sets by progressively downscaling the images to 1600 × 1200, 1200 × 900, 800 × 600 and 400 × 300. Then, we randomly subdivided each data set in 5 sets of 20 images which have been attacked by means of CLBA with removal rate {50, 60, 70, 80, 100} respectively. From the ROC curves of Fig. 9 it turns out that the performances of KCR and SVM do not depend excessively on the image size, as opposed to CHI, for which the detection is easier on larger images. We investigated more in-depth the relationship between downscaling and keypoint removal detection by considering the sizes 1600 × 1200 and 400 × 300. On average, the reductions of keypoints and corners due to resizing are consistent with each other (81% and 83%) and consequently the corresponding KCR index is not significantly altered. On the contrary, following resizing the proportion of *medium* variance blocks undergoes a variation of about 12% with respect to the total amount of blocks, causing erroneous assignments to the class of *low* variance blocks; this variation reduces the performance of the CHI detector based on the *medium* reference histograms. The SVM detector, which relies on all the classes of variance, does not seem to be affected by such a problem.

To determine the thresholds, we fixed a maximum value of 0.1 for the false positive rate and we gathered the corresponding value for the true positive rate from each curve. Finally, we retrieved the threshold responsible for that specific point in the ROC curves. The results summarised in Tab. III show that the KCR threshold is stable for the various sizes, whereas the CHI and SVM thresholds require more tweaking.

TABLE III
THRESHOLDS VS IMAGE SIZE FOR A PROBABILITY OF FALSE ALARM 0.1

Method	400 × 300	800 × 600	1200 × 900	1600 × 1200
KCR	−1.9	−1.9	−1.9	−1.9
CHI	48	32	22	20
SVM	0.41	0.42	0.58	0.67

F. Detection of Local Keypoint Removal

In practical applications, the attacker may not need to remove all the SIFT keypoints (or a large portion of them) in an image, but rather just a small portion according to the forgery. This may be the case, for example, of an attacker trying to impair a SIFT-based copy-move detector (see Sec. VII for details). For this reason, we now take into consideration the case in which keypoints are removed from a specific area of the image.

To reveal a local forgery, we apply the detectors in a blockwise fashion: each 32×32 non-overlapping block of the image is processed by analysing the statistics of a larger square region surrounding it. In fact, a statistical analysis of a small block would not be meaningful enough; consequently, on the one hand the analysed region should be large enough, while on the other hand it should be small enough so that its characteristics are representative of the to-be-classified block.

In practice, for each block we run the detectors on the 600×600 area surrounding it, we compute a soft value describing the degree of tampering and we assign it to the inner 32×32 block. Then, we shift by 32 pixels and we repeat the procedure. This procedure provides a soft-valued map roughly localising areas artificially deprived of SIFT keypoints, which is binarised by applying the detector's threshold. The map is finally cleaned by removing those regions whose area is smaller than 2% of the total image area.

The results we show in this section have been obtained by using a 600×600 region to classify each image block but, according to our experiments, such a size can be reduced to 300×300 pixels at the cost of an increased processing time. Localisation, in fact, is computationally intense as it takes, for example, about 9 minutes with a 600×600 window on a 64 bit OS with 8 GB RAM to process the 1333×2000 image of Fig. 10 (top). The main advantage of a smaller window is the capability of revealing tampered regions of minimum size in the order of 300×300 pixels. In this latter case, the average percentage of keypoints belonging to a patch with respect to number of keypoints of the whole image is around 4.1%. Such a percentage is satisfactory and, above all, in line with the localisation resolution performances shown by SIFT-based methods designed to detect copy-move forgeries.

Fig. 10 displays an example of local removal detection by means of KCR (T_1=-1.9). The keypoints of an authentic image were removed from the framed region (top left), with decreasing removal rate of 100%, 80%, 60%, 40% and 20%. As expected, detection becomes more difficult as the removal rate lowers; nevertheless, even in the case of the lowest removal rate, the detector is still able to correctly localise 39% of the manipulated area.

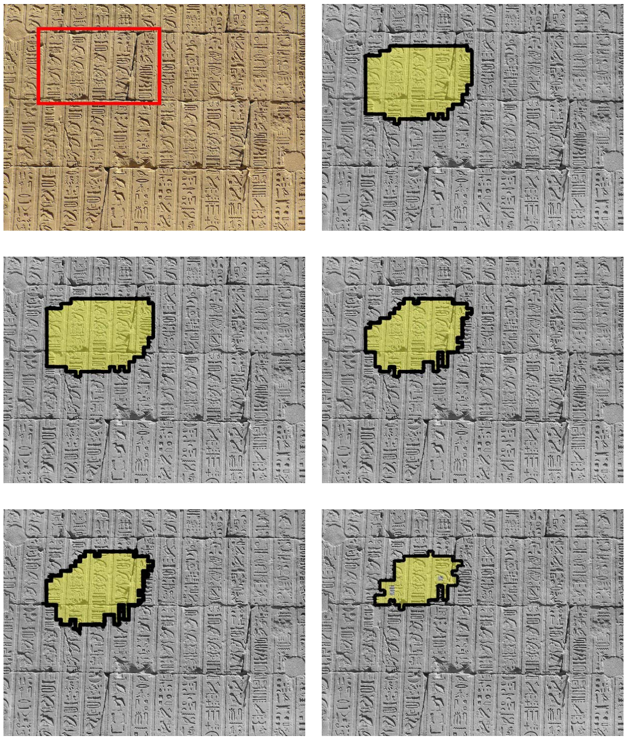


Fig. 10. Example of local removal. From top left to bottom right: authentic image; 100%, 80%, 60%, 40% and 20% removal rate.

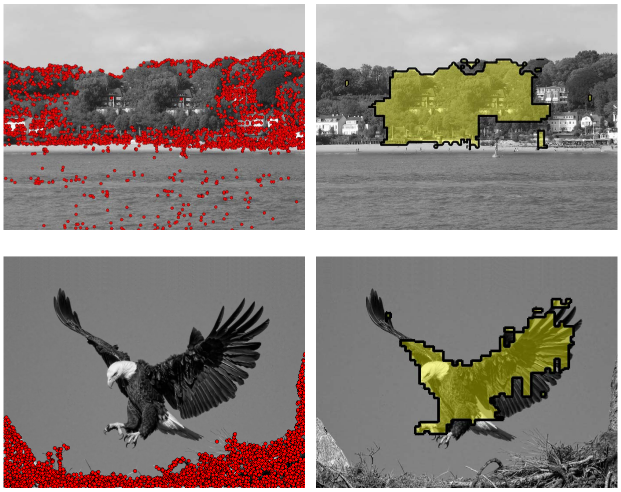


Fig. 11. Local removal in presence of authentic regions with few or no keypoints. Left: keypoints following CLBA; right: masking avoids false positives in the sea and in the sky.

One issue with the windowed approach is that blocks surrounded by regions naturally poor of keypoints, such as in the sky or sea areas, are erroneously considered tampered. Our solution conveniently resorts to the binarised variance map V: first, very low blocks (variance ≤ 0.1) are set to 0; then, morphological flood-fill and area opening are used to remove holes and isolated pixels. The actual tampering map is obtained by AND-ing V and the localisation map provided by the removal detector. Two examples of this procedure are shown in Fig. 11.

V. SIFT Keypoint Injection

Arguably, the most straightforward strategy to bypass keypoint removal detection consists in introducing *fake* keypoints in suitably chosen positions. If enough of such keypoints are reintroduced, then the discriminative power of the forensic tools of Sec. III may be compromised. This is the typical case of SIFT-based copy-move counter-forensics, whereby a wise attacker who has removed the matching keypoints revealing the forgery injects fake keypoints to bypass the localisation by means of the proposed detectors (we will explore this scenario in Sec. VII).

To further investigate this topic, we devised a new injection attack based on the exploratory studies presented in [13]. In [13], it has been proved that image enhancement tools can effectively inject keypoints by exalting image details. In particular, the following algorithms have been examined: Anisotropic Diffusion [23], Contrast Limited Adaptive Histogram Equalisation (CLAHE) [24], Brightness Preserving Fuzzy Histogram Enhancement (BPFHE) [25] and Gaussian Smoothing.[3] To these four enhancers, we added the Forging with Minimum Distortion (FMD) attack proposed by Do *et al.* in [10]; such attack triggers false positives at the SIFT contrast check by adding a patch (obtained by minimising the local distortion) to the neighbourhood of the injection location.

The advantage of FMD with respect to the use of image enhancers is that it allows to choose the coordinates at which a keypoint should be injected. Consequently, with FMD we can define suitable locations wherein to inject the keypoints, while with the rest of the tools keypoints are injected randomly and the appropriateness of their positions verified afterwards. On the other hand, FMD strongly affects the quality of the forgery and thus it should be used sparingly.

The attack we used in this paper resorts to three of the above image enhancers and to the FMD with an approach that is similar to CLBA, that is a classification-driven procedure exploiting the advantages of each single algorithm while limiting its weaknesses. For this reason we refer to it as Classification-Based Injection (CLBI). Intuitively, CLBI injects keypoints with FMD where they *must* be found (i.e. in proximity of corners and edges) and with the three enhancers where they *should* be found (i.e. in sufficiently textured, non-flat regions). Therefore, the classification task basically consists in distinguishing between regions containing edges and corners and the rest of the image.

A. Scheme of the Proposed Injection Method

CLBI takes as input an image I_{rem} without keypoints and produces the injected image J in four steps as shown by the block diagram in Fig. 12: 1) classification of I_{rem}'s regions (red blocks); 2) FMD injection (green); 3) contrast-enhancement injection (orange); 4) match refinement (cyan). In the sequel we provide details on each block.

1) Region Classification: To distinguish image regions according to the structures they contain, we apply the tensor operator [23] to I_{rem}, producing a map of the same size of the image highlighting edges and flow-like, T-shaped and Y-shaped structures. By normalising the output of the operator in $[0, 1]$, we obtain the injection map Map wherein each element (i, j) quantifies the saliency of the corresponding pixel. The idea is to use the most salient points such that $Map(i, j) \geq 0.5$ as preferred injection locations for the FMD, while the remaining less descriptive points (whose score is however not null) are left for the image enhancers.

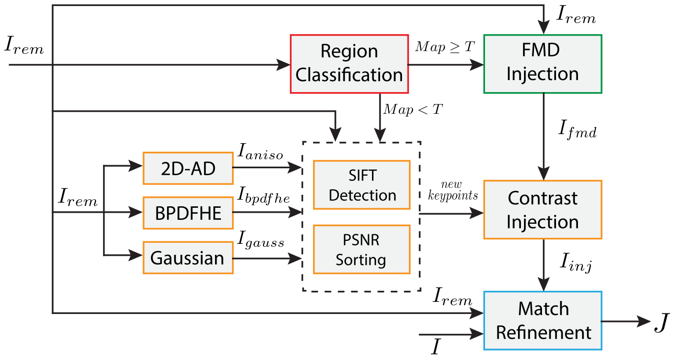


Fig. 12. Schematisation of the injection framework. Keypoints are injected with FMD or contrast enhancement tools depending on the saliency of the image regions and then refined to avoid matches with the original image.

2) FMD Injection: The coordinates of all the pixels belonging to the salient regions as well as the input image I_{rem} are fed to the first injection block, where the FMD attempts to introduce a keypoint in each location. The large amount of candidate locations is a direct consequence of FMD's characteristics: the artefacts it introduces are visually unacceptable if the attack is too intense, thus forcing us to set its strength to the minimum (that is 1); by doing so, however, the attack can not ensure a positive outcome and for this reason we repeat the injection for all the locations; following each attempt, we run a SIFT check to verify whether a new keypoint has been introduced. If this is the case, we do not further alter its 8×8 neighbourhood, to avoid undoing the forgery. This stage produces a first version I_{fmd} of the injected image.

3) Contrast-Enhancement Injection: A copy of the input image I_{rem} is fed to each enhancer to produce the three partially injected images referred to as I_{aniso}, I_{bpdfhe} and I_{gauss} in Fig. 12. All the 8×8 neighbourhoods of the injected keypoints of these images are extracted and ordered according to their local PSNR with respect to I_{rem}. If more than one enhancer has created a keypoint in the same location, we keep only the one with highest PSNR. The goal of this procedure is to generate as much keypoints of acceptable quality as possible to repopulate the more uniform (but still significant) regions of I_{fmd}, which were not touched by the previous stage of the attack; therefore, keypoints passing this selection are pasted with their neighbourhoods into I_{fmd} only if $0 < Map(i, j) < 0.5$, producing the injected image I_{inj}.

4) Match Refinement: Finally, we check that the keypoints of I_{inj} do not match with their counterparts in the authentic image. We consider a match correct if the distance between the old and new descriptors falls below a certain threshold

[3] It has been observed experimentally in [10] that Gaussian blur with a suitably large width $\sigma > 2$ tends to create rather than remove keypoints.

(8 in our case). This final stage addresses two types of matches at the same time, i.e. those accidentally created by the injection procedure and those left because of non-perfect removal. We suppress the former by restoring the corresponding neighbourhood from I_{rem}. If such neighbourhood already contains keypoints, then they are suppressed by applying the Removal with Minimum Distortion (RMD) attack [10]; in fact, RMD is very effective when its strength δ is increased at least to 3 (against a default of 1), although this tends to introduce salt and pepper noise.

The output of this stage is the final refined image J. It is worth noting that, despite all the controls, J may still have some correct matches, mainly due to two factors: the changes in pixel values while pasting overlapping neighbourhoods and the extreme robustness of some keypoints even to the strengthened RMD. The number of such matches is very small when compared to that of the authentic ones (see Sec. VI-A and Fig. 14 in particular).

VI. Experimental Validation of Keypoint Injection

We now investigate the robustness of the proposed keypoint removal detectors against injection. First, we study CLBI's performance in terms of injected keypoints and correct matches between the forgery and the authentic image and then we evaluate whether it represents a threat to the detectors.

A. Performance of Keypoint Injection

In order to keep the complexity of the experiments under control,[4] we considered smaller images, i.e. those composing the UCID data set [26], which consists of 1338 images of size 512 × 384 pixels. Concerning the experimental setup, the parameters of each injection algorithm needed by CLBI were set to their default values as in [23]–[25], and descriptors were matched by means of nearest neighbour (threshold 0.8 as in [3]) and k-d trees [27]. Given the similarity of the results, we discuss only the former matching strategy.

1) Injected Keypoints: We quantified the injection effectiveness by means of the Keypoint Injection Rate (KIR), that is to say the percentage of forged keypoints with respect to the original amount prior to keypoint removal. First, we removed the keypoints from each image; secondly, we forged new keypoints with CLBI and we computed the KIR; finally, we organised all the KIRs in a histogram. We show the envelope of such histogram in Fig. 13 and we compare it to those of the class-unaware injection algorithms presented in [13]. CLBI provided an average KIR of 49.7% outperforming the rest of the algorithms, which attained the following rates: 27.9% for Gaussian Smoothing; 23.4% for Anisotropic Diffusion; 17% for FMD; 14% for CLAHE; and 11% for BPFHE.

2) Matches Among Authentic Images and Forgeries: In Fig. 14 we show the cumulative distribution (with superposed normalised histogram) of the matches among authentic and forged images after removal (dash-dotted blue line) and after injection (solid red line). Following refinement, the matches left because of non-perfect removal and those introduced accidentally during injection are effectively reduced, to the point that 61% of the images have less than 3 matches (versus 39% before injection); furthermore, the images without matches increased from 11% to 25% of the data set. Such results are satisfactory, especially if we consider that prior to the attack images have on average 232 matches.

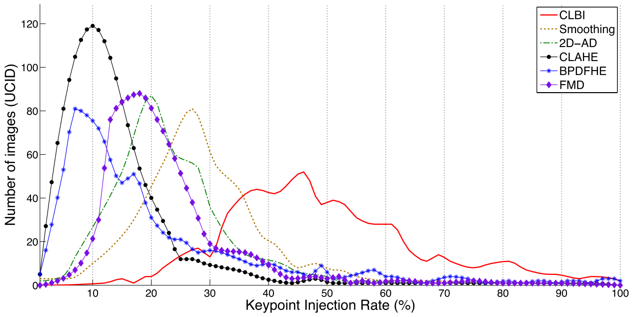


Fig. 13. KIR envelopes for the CLBI attack and for the class-unaware injection algorithms presented in [13].

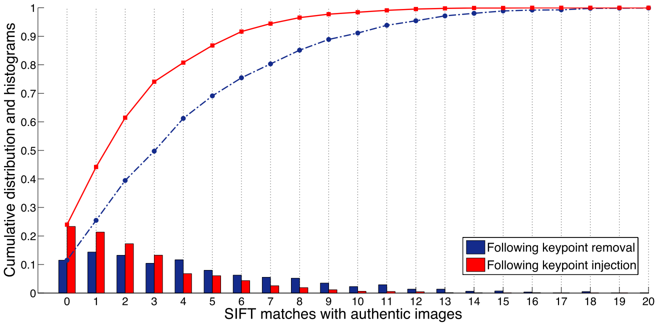


Fig. 14. Cumulative distribution (with superposed normalised histogram) of correct matches following removal (dash-dotted blue line) and following injection (solid red line).

Even though the number of injected keypoints may appear low when compared to the original ones, it should be considered that this is a consequence of our assumptions rather than a limitation of the algorithm. In fact, we imposed that no matches should be found between the authentic and forged image in order for the method to be practically useful against copy-move detection. In those applications where such a constraint can be relaxed, the injection algorithm is capable of creating up to 82% of the original amount of keypoints, with 16 average matches.

3) Visual Quality of the Forgery: In Tab. IV we give some results about the quality loss caused by removal and injection in terms of PSNR and SSIM, evaluated both globally (averaged on all the data set) and locally (averaged on all the attacked neighbourhoods). Both the visual quality with respect to the authentic image and to the outcome of removal are not significantly worsened by the injection. We close this series of experiments with Fig. 15, where we show the keypoints of two authentic images (left) and their corresponding forgeries (right).

In conclusion, the proposed injection attack allows to reach a good compromise between percentage of removed keypoints,

[4] On average, our Matlab implementation takes up to 180 seconds on a 400 × 400 image, due to the several thousands of iterations required by the FMD, each of which also includes SIFT detection (64 bit OS, 8 GB RAM).

TABLE IV

QUALITY OF REMOVAL-INJECTION: AVERAGE PSNR (db) AND SSIM ON ENTIRE IMAGES AND ON THE ATTACKED NEIGHBOURHOODS

Manipulation	Global quality		Local quality	
	PSNR	SSIM	PSNR	SSIM
After removal	32.74	0.976	28.88	0.894
After injection	30.98	0.97	27.29	0.839

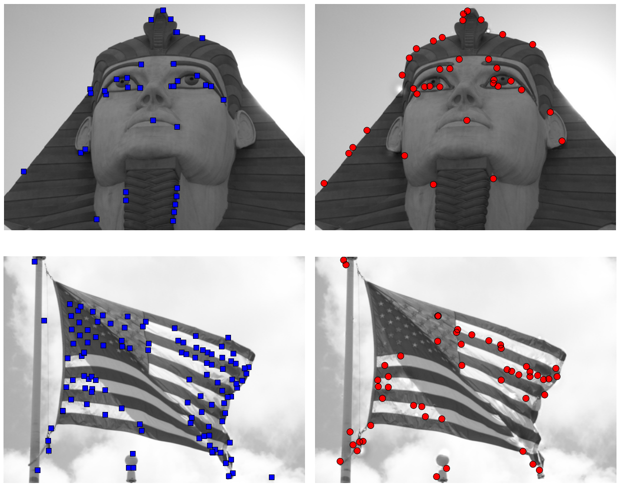


Fig. 15. Original (left) and injected keypoints (right). The 46 (132) authentic keypoints were removed by CLBA and replaced with 40 (58) fakes by CLBI.

percentage of injected keypoints, correct matches accidentally left or introduced and perceptual quality of the forged image. In the next section we will evaluate whether such tool is powerful enough to counter keypoint removal detection.

B. Impact of Injection on Keypoint Removal Detection

We used 300 images to assess the effect of injection on the detectors of Sec. III: 100 randomly drawn from the UCID data set; the corresponding 100 CLBA forgeries (effective removal rate $\geq$ 90%); the same 100 images after CLBI injection. We ran each of the detectors on all the images, we collected the detection scores and we organised them into scattergrams like in Sec. IV-D. In Fig. 16 we show the scattergrams relative to the KCR (top) and SVM (bottom) detectors. The blue squares represent authentic images, red circles and green triangles correspond to images after keypoint removal and keypoint removal-injection. For sake of brevity, we omit CHI's scattergram because its trend is similar to that of the SVM.

The capability of separating the classes of authentic and CLBA-forged images is in line with the results obtained with different data sets in Sec. IV-D, but the scores of injected images tend to scatter and mix with those of the other two classes. This phenomenon is noticeable especially for SVM and CHI detectors, while KCR proves once again to be the most robust tool, as green triangles still appear separable from blue squares despite their proximity (see Fig. 16). This observation is corroborated by the data of Tab. V showing the confusion matrices for each of the detectors when the thresholds suggested in Sec. IV-E for small images are used ($T_1 = -1.9$, $T_2 = 48$ and $T_3 = 0.41$). Authentic and CLBA-forged images are correctly labelled with average detection accuracies of 92.5%, 93.5% and 100%, whereas the accuracy in labelling injected images as forgeries is 84%, 2% and 26% respectively; the consequences of keypoint injection are severe on CHI and SVM, for which slightly better results can be obtained, at the expense of the capability of identifying authentic images, by tweaking T_2 and T_3.

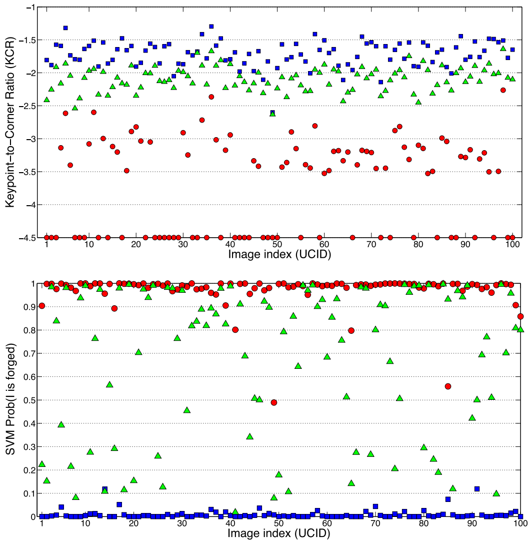


Fig. 16. KCR (top) and SVM (bottom) scattergrams; blue squares: authentic images; red circles: CLBA forgeries; green triangles: injected images.

TABLE V

CONFUSION MATRICES BEFORE AND AFTER INJECTION FOR THE PROPOSED DETECTORS

	KCR ($T_1 = -1.9$)		**CHI** ($T_2 = 48$)		**SVM** ($T_3 = 0.41$)	
	Authentic	Forged	Authentic	Forged	Authentic	Forged
Authentic	0.85	0.15	0.98	0.02	1	0
CLBA	0	1	0.11	0.89	0	1
Injected	0.16	0.84	0.98	0.02	0.74	0.26

TABLE VI

DETECTION ACCURACY FOR THE 3-CLASS SVM DETECTOR

	Authentic	CLBA	Injected
Authentic	0.83	0.02	0.15
CLBA	0.01	0.94	0.04
Injected	0.15	0.05	0.8

C. Three-Class SVM Detector

Based on the above observations, we improved the SVM detector by reformulating the underlying classification as a 3-class problem, in such a way to discriminate between authentic, CLBA-forged and injected images. In doing so, not only we were able to recognise a forged image but also to identify the manipulation it has undergone.

We have used 1200 UCID images of each class to build the probabilistic SVM model (200 for 5-fold cross-validation, 1000 for training with $C = 32$ and $\gamma = 0.125$) and the remaining 138 images per class to test it. In Tab. VI we report the confusion matrix we obtained by assigning each test image to the class corresponding to the maximum of the output probability. The discriminative power of the 3-class SVM in presence of keypoint injection is now comparable to that of the KCR, with authentic and injected images misclassified in the 15% of the cases. We believe that such a behaviour is at least in part due to the nature of the UCID data set, whose relatively low amount of keypoints in images makes distinctions harder, and that it should subside as the image size grows.

VII. Application to Copy-Move Detection

As a final test, we investigate the interplay between all the tools described so far in the context of the detection of copy-move forgeries; this is, in fact, a typical image forensic problem that can be effectively tackled with by relying on SIFT features, whose robustness and distinctiveness allow to reliably match cloned areas. The goal of copy-move counter-forensics is to create a forgery that is undetectable by SIFT-based techniques like [4] or [5]; a wise counterfeiter can attain such a goal by first applying keypoint removal to disable the targeted algorithm and then keypoint injection to hide the traces of removal exploited by our detectors.[5] It goes without saying that all the manipulations are carried out locally on one (or more) of the cloned areas and leave the rest of the image unaltered.

A. Evaluation Procedure and Employed Detectors

Without loss of generality, we considered only two copy-moved areas, the source A_1 and its clone A_2, both containing a fair amount of keypoints because otherwise concealing the forgery would be trivial. We collected 10 images ranging from 1600×1200 to 3000×2000 pixels (some belonging to the data set used in [2]) and we created 10 realistic forgeries by duplicating one region of size in the order of 300×400 pixels and variable shape (see for example Figs. 17 and 19 (a)–(b)).

We chose Amerini *et al.*'s copy-move forgery detector (CMFD) [29], which improves the one presented by the same authors in [5]. In a nutshell, following SIFT feature detection and hierarchical clustering, the algorithm in [5] considers an image as forged if at least 3 matches are found within pairs of image regions. The major drawback with this approach is that requiring only 3 matches may lead to a large number of false positives, especially in images with many keypoints and repeated texture patterns such as walls. To overcome this problem, a new clustering technique was devised in [29] allowing to better estimate the affine transformation between two sets of matched points. By extending such transformation to the image regions underlying matching sets, it becomes possible to localise the tampered areas. According to the improved detector, an image is tampered if at least one affine transformation is found among pairs of clusters. We refer to [5] and [29] for a detailed description of the algorithms. We detected keypoint removal by means of the KCR detector which, based on the experiments of Secs. IV and VI, proved to be the most robust detector.

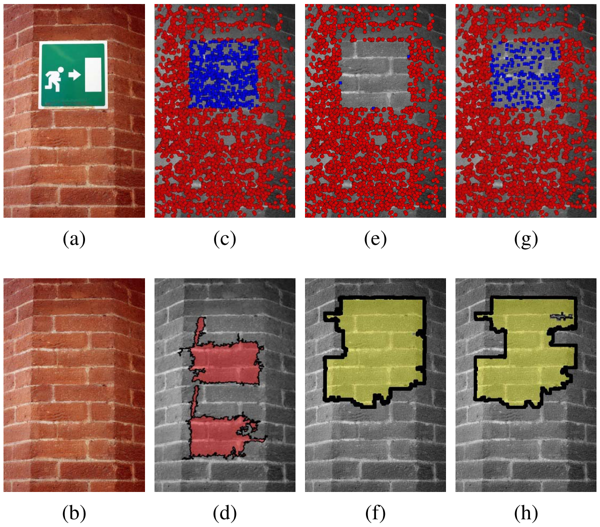


Fig. 17. Copy-moved image I_2. (a) Authentic; (b) copy-move forgery; (c) keypoints of (a) (blue square markers indicate the keypoints of the cloned area); (d) output of the CMFD; (e) keypoints following removal; (f) output of KCR on (e); (g) keypoints following injection; (h) output of KCR on (g).

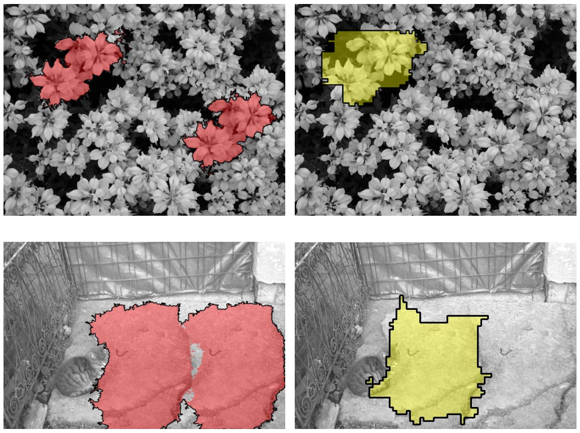


Fig. 18. Examples of localisation. Left: CMFD output highlighting copy-moved regions; right: KCR output following removal.

Since the majority of keypoints are matching across the cloned areas, we let the attacker remove all the keypoints of A_2 by means of CLBA; those matches that are robust enough to survive the attack on A_2 are attacked on A_1. Then, A_2 is repopulated by means of CLBI. This solution limits the perceptibility of the attacks by taking advantage of the fact that a match can be deleted by removing only one of its members.

B. Experimental Analysis

Following each stage of the manipulation, the performance of the CMFD and the KCR detector are measured at image level to evaluate their capability of detecting tampered images and at the pixel level to assess their accuracy in localising forged regions. For the first level, we consider the fraction of tampered images that are correctly identified. The metrics we

[5] Block-based detection is not considered; however, due to its lack of robustness to geometric manipulations, it can be disabled by cascading CLBA and simple geometric attacks such as the one in [28], as shown in [7].

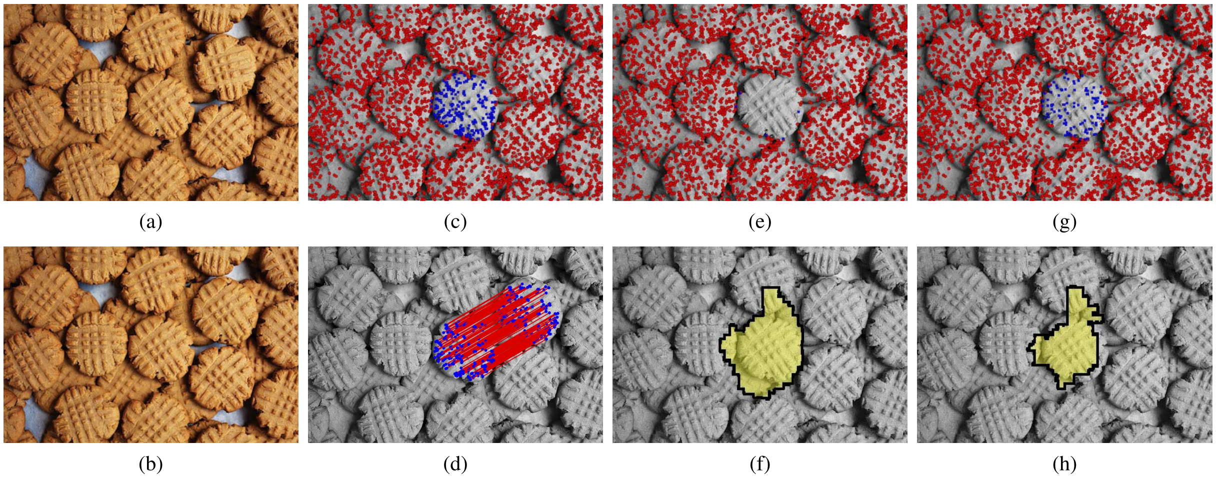


Fig. 19. Copy-moved image I_3. (a) Authentic; (b) copy-move; (c) keypoints of (a) (blue square markers indicate the keypoints of the cloned area); (d) matches revealed by the CMFD before keypoint removal; (e) keypoints following removal; (f) output of KCR on (e); (g) keypoints following injection; (h) output of KCR on (g).

TABLE VII

KEYPOINTS IN A_2, MATCHES BETWEEN A_1 AND A_2 AND BINARY DECISION ON AUTHENTICITY ACCORDING TO THE CMFD

	PLAIN COPY-MOVE			AFTER REMOVAL			AFTER INJECTION		
Img	keypoints	matches	forged	keypoints	matches	forged	keypoints	matches	forged
I_1	157	130	1	1	1	0	85	0	0
I_2	1023	868	1	5	0	0	321	1	0
I_3	284	214	1	6	1	0	126	0	0
I_4	516	417	1	1	1	0	128	2	0
I_5	182	88	1	1	0	0	44	0	0
I_6	782	579	1	11	1	0	208	0	0
I_7	1211	929	1	2	0	0	184	2	0
I_8	1221	989	1	28	0	0	367	0	0
I_9	597	458	1	10	0	0	263	0	0
I_{10}	306	152	1	0	0	0	87	0	0

chose for the second level are precision p and recall r:

$$p = \frac{Tp}{Tp + Fp} \qquad r = \frac{Tp}{Tp + Fn}. \tag{6}$$

When (6) are applied on a pixel basis, Tp is the number of forged pixels that are correctly identified; Fp is the number of authentic pixels erroneously labelled as forged; Fn is the number of forged pixels erroneously labelled as authentic. Hence, precision is the fraction of pixels identified as tampered that are truly tampered and recall (or true positive rate) is the fraction of tampered pixels that are correctly classified as such. Precision and recall can be conveniently combined by considering their harmonic mean, called F_1-score, as follows:

$$F_1 = 2\,\frac{p \cdot r}{p + r}. \tag{7}$$

1) Authenticity Verification: In our implementation, an image is forged according to the CMFD if there is at least one affine transformation linking A_1 to A_2, and according to the KCR detector if there is at least one tampered region whose area is $\geq 2\%$ of the image area. In Tab. VII, for each test image we report the number of keypoints of A_2, matches between A_1 and A_2 and the binary decision on image authenticity according to the CMFD in the plain copy-move, following removal and following removal-injection. Regardless of the underlying detection criterion (at least either 3 matches as in [5] or 1 affine transformation as in [29]), the CMFD reveals 100% of the plain forgeries but is always disabled by the keypoint manipulations. Obviously, the KCR detector is incapable of discriminating until keypoints are actually altered; in the other cases, it recognises 100% of the counter-forensically treated copy-moves, since the regions it detects have on average an area which is 12% of the total image area after removal and 9% after removal-injection.

TABLE VIII

PRECISION, RECALL AND F_1-SCORE FOR THE CFMD AND THE KCR

	PLAIN COPY-MOVE			AFTER REMOVAL			AFTER INJECTION		
	p	r	F_1	p	r	F_1	p	r	F_1
CMFD	0.84	0.67	0.75	0	0	0	0	0	0
KCR	0	0	0	0.6	0.85	0.7	0.74	0.67	0.7

2) Tampering Localisation: Tab. VIII shows the localisation accuracy of the CMFD and the KCR in terms of average p, r and F_1-score (computed over all the images' pixels) following each stage of the forgery and confirms the inadequateness of the CMFD in presence of keypoint manipulations. The repercussion of injection on KCR's performance is twofold: on the one hand, the probability of detecting forged pixels (i.e. r) is lowered by 18%, even though the reduction is insufficient to hide keypoint removal; on the other hand, the false positives caused by the sliding window on the borders of the tampered areas are also reduced, hence explaining the higher p.

Figs. 17 (I_2) and 19 (I_3), wherein the exit sign was hidden by replicating a portion of the wall and a cookie was duplicated, exemplify well the data of Tabs. VII–VIII and the capability of injection to hide the traces of keypoint removal to a visual investigation. The distributions of SIFT keypoints in Figs. 17–19 (e) can still raise the suspicion of a keen observer but those of Figs. 17–19 (g) certainly require specialised tools.

It is also worth noting that the CMFD allows to localise both the cloned areas whilst the KCR detector, due to the attacker's strategy, can identify only the one that has been tampered with. Two additional examples of localisation are shown in Fig. 18: the left column corresponds to the CMFD output on the plain copy-moves and the right column to the KCR output on the CLBA-attacked forgeries (on which the CMFD fails).

It must also be said that adversary-unaware forensic methods could detect and localise the regions altered by means of keypoint removal and injection. Methods which examine the statistical consistency over the whole image, such as for example the one proposed by Pan *et al.* in [30], could reveal anomalies due to the cancellation of salient features. However, it is out of the scope of this paper to make a comparison among adversary-aware/unaware techniques to detect this kind of alterations.

In conclusion, we can say that in the examined scenario the adversary fails to evade copy-move detection if the forensic analyst resorts to the combination of the two above categories of detectors, e.g. by OR-ing their binary outputs on image authenticity or the tampering maps.

VIII. Conclusion

We tackled with the growing attention given by counter-forensic research on deceiving SIFT-based copy-move detection techniques. Because the existing attacks against such techniques delete SIFT keypoints to suppress the matches linking the cloned areas, we have devised three forensic detectors revealing global or local keypoint removal, based on the anomalies of the distribution of keypoints after the manipulation. The detectors are effective not only against keypoint removal but also against the injection of fake keypoints as a means to conceal removal, as confirmed by the results we obtained in supporting a state-of-the-art copy-move detector that was disabled by the above two forgeries.

Among the open issues, we mention: investigating the possibility of recognising the injection forgery by studying potential anomalies in the properties of the fake keypoints with respect to the original ones (e.g. distribution of scales or dominant orientations); extending keypoint removal attacks to the higher octaves to further assess the effectiveness of the newly devised detectors. Such attacks, in fact, would allow to counter those applications that, unlike copy-move detection, rely on less numerous but more robust keypoints.

References

[1] R. Böhme and M. Kirchner, "Counter-forensics: Attacking image forensics," in *Digital Image Forensics*, H. T. Sencar and N. Memon, Eds. New York, NY, USA: Springer-Verlag, 2012, ch. 10.

[2] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou, "An evaluation of popular copy-move forgery detection approaches," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 6, pp. 1841–1854, Dec. 2012.

[3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[4] X. Pan and S. Lyu, "Region duplication detection using image feature matching," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 857–867, Dec. 2010.

[5] I. Amerini, L. Ballan, R. Caldelli, A. D. Bimbo, and G. Serra, "A SIFT-based forensic method for copy–move attack detection and transformation recovery," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 1099–1110, Sep. 2011.

[6] R. Caldelli, I. Amerini, L. Ballan, G. Serra, M. Barni, and A. Costanzo, "On the effectiveness of local warping against SIFT-based copy-move detection," in *Proc. 5th Int. Symp. Commun., Control, Signal Process. (ISCCSP)*, Rome, Italy, May 2012, pp. 1–5.

[7] I. Amerini, M. Barni, R. Caldelli, and A. Costanzo, "Counter-forensics of SIFT-based copy-move detection by means of keypoint classification," *EURASIP J. Image Video Process.*, vol. 2013, no. 1, p. 18, 2013.

[8] C.-Y. Hsu, C.-S. Lu, and S.-C. Pei, "Secure and robust SIFT," in *Proc. 17th ACM Int. Conf. Multimedia (MM)*, New York, NY, USA, 2009, pp. 637–640.

[9] T.-T. Do, E. Kijak, T. Furon, and L. Amsaleg, "Understanding the security and robustness of SIFT," in *Proc. 18th ACM Int. Conf. Multimedia*. New York, NY, USA, 2010, pp. 1195–1198.

[10] T.-T. Do, E. Kijak, T. Furon, and L. Amsaleg, "Deluding image recognition in SIFT-based CBIR systems," in *Proc. 2nd ACM Workshop Multimedia Forensics, Security, Intell.*, New York, NY, USA, 2010, pp. 7–12.

[11] T.-T. Do, E. Kijak, L. Amsaleg, and T. Furon, "Enlarging hacker's toolbox: Deluding image recognition by attacking keypoint orientations," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Mar. 2012, pp. 1817–1820.

[12] C.-S. Lu and C.-Y. Hsu, "Constraint-optimized keypoint inhibition/insertion attack: Security threat to scale-space image feature extraction," in *Proc. 20th ACM Int. Conf. Multimedia*, 2012, pp. 629–638.

[13] I. Amerini, M. Barni, R. Caldelli, and A. Costanzo, "SIFT keypoint removal and injection for countering matching-based image forensics," in *Proc. 1st ACM Workshop Inform. Hiding Multimedia Security (IH MMSec)*, 2013, pp. 123–130.

[14] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[15] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, vol. 15. Manchester, U.K., 1988, p. 50.

[16] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 1994, pp. 593–600.

[17] O. Pele and M. Werman, "The quadratic-chi histogram distance family," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2010, pp. 749–762.

[18] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advance in Large Margin Classifiers*. Cambridge, MA, USA: MIT Press, 1999, pp. 61–74.

[19] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," in *Proc. ACM Int. Conf. Multimedia*, 2010, pp. 1469–1472. [Online]. Available: http://www.vlfeat.org

[20] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2008, pp. 304–317. [Online]. Available: http://lear.inrialpes.fr/ jegou/data.php

[21] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2005, pp. 1508–1511.

[22] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, 2011. [Online]. Available: http://www.csie.ntu.edu.tw/cjlin/libsvm/

[23] J. Weickert and H. Scharr, "A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance," *J. Vis. Commun. Image Represent.*, vol. 13, nos. 1–2, pp. 103–118, 2002.

[24] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics Gems IV*. New York, NY, USA: Academic, 1994, pp. 474–485.

[25] D. Sheet, H. Garud, A. Suveer, M. Mahadevappa, and J. Chatterjee, "Brightness preserving dynamic fuzzy histogram equalization," *IEEE Trans. Consum. Electron.*, vol. 56, no. 4, pp. 2475–2480, Nov. 2010.

[26] G. Schaefer and M. Stich, "UCID: An uncompressed color image database," in *Proc. Electron. Imag. Int. Soc. Opt. Photon.*, 2003, pp. 472–480. [Online]. Available: http://homepages.lboro.ac.uk/ cogs/datasets/ucid/ucid.html

[27] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[28] H. C. Nguyen and S. Katzenbeisser, "Security of copy-move forgery detection techniques," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2011, pp. 1864–1867.

[29] I. Amerini, L. Ballan, R. Caldelli, A. D. Bimbo, L. D. Tongo, and G. Serra, "Copy-move forgery detection and localization by means of robust clustering with J-linkage," *Signal Process., Image Commun.*, vol. 28, no. 6, pp. 659–669, 2013.

[30] X. Pan, X. Zhang, and S. Lyu, "Exposing image forgery with blind noise estimation," in *Proc. 13th ACM Multimedia Workshop Multimedia Security*, 2011, pp. 15–20.

Andrea Costanzo received the Laurea degree in telecommunications engineering and the Ph.D. degree in information engineering from the University of Siena, Siena, Italy, in 2009 and 2014, respectively. He is a member of the Visual Information Processing and Protection Group at the Department of Information Engineering and Mathematics, University of Siena, and the National Inter-University Consortium for Telecommunications. His main research interests focus on multimedia forensics and counter forensics.

Irene Amerini received the Laurea degree in computer engineering and the Ph.D. degree in computer engineering, multimedia, and telecommunication from the University of Florence, Florence, Italy, in 2006 and 2010, respectively. She is currently a Post-Doctoral Researcher with the Image and Communication Laboratory, Media Integration and Communication Center, University of Florence. She was a Visiting Scholar with Binghamton University, Binghamton, NY, USA, in 2010. Her main research interests focus on multimedia content security technologies, secure media, and digital and multimedia forensics.

Roberto Caldelli (M'11) received the degree in electronic engineering and the Ph.D. degree in computer science and telecommunication from the University of Florence, Florence, Italy, in 1997 and 2001, respectively.

From 2005 to 2013, he was an Assistant Professor with the Media Integration and Communication Center, University of Florence. In 2014, he joined the National Inter-University Consortium for Telecommunications (CNIT) as a Permanent Researcher. His main research activities, witnessed by several publications, include digital image processing, interactive television, image and video digital watermarking, and multimedia forensics. He holds two patents in the field of content security and multimedia interaction.

Mauro Barni (M'92–SM'06–F'12) received the degree in electronic engineering and the Ph.D. degree in informatics and telecommunications from the University of Florence, Florence, Italy, in 1991 and 1995, respectively. He is currently an Associate Professor with the University of Siena, Siena, Italy. During the last decade, his activity has focused on digital image processing and information security, with a particular reference to the application of image processing techniques to copyright protection (digital watermarking) and multimedia forensics. Recently, he has been studying the possibility of processing signals that have been previously encrypted without decrypting them. He led several national and international research projects on these subjects. He has authored about 270 papers, and holds four patents in the field of digital watermarking and document protection. He has coauthored the book *Watermarking Systems Engineering* (Dekker, 2004). He was a recipient of the IEEE SIGNAL PROCESSING MAGAZINE Best Column Award in 2008, and the IEEE Geoscience and Remote Sensing Society Transactions Prize Paper Award in 2011. He was the Chairman of the IEEE Multimedia Signal Processing Workshop (Siena, 2004), and the International Workshop on Digital Watermarking (Siena, 2005). He was the founding Editor-in-Chief of the *EURASIP Journal on Information Security*. He currently serves as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY. From 2010 to 2011, he 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 was appointed as a Distinguished Lecturer of the IEEE Signal Processing Society from 2012 to 2013.