SIFT Keypoint Removal and Injection via Convex Relaxation

Yuanman Li, *Student Member, IEEE,* Jiantao Zhou, *Member, IEEE,* An Cheng, Xianming Liu, *Member, IEEE,* and Yuan Yan Tang, *Fellow, IEEE*

***Abstract*—*Scale Invariant Feature Transform* (SIFT), as one of the most popular local feature extraction algorithms, has been widely employed in many computer vision and multimedia security applications. Although SIFT has been extensively investigated from various perspectives, its security against malicious attacks has rarely been discussed. In this work, we show that the SIFT keypoints can be effectively removed with minimized distortion on the processed image. The SIFT keypoint removal is formulated as a constrained optimization problem, where the constraints are carefully designed to suppress the existence of local extrema and prevent generating new keypoints within a local cuboid in the scale space. To hide the traces of performing SIFT keypoint removal, we then propose to inject a large number of fake SIFT keypoints into the previously cleaned image with minimized distortion. As demonstrated experimentally, our proposed SIFT removal and injection algorithms significantly outperform the state-of-the-art techniques. Further, it is shown that the combined SIFT keypoint removal and injection attack strategy is capable of defeating the most powerful forensic detector designed for SIFT keypoint removal. Our results suggest that an authorization mechanism is required for SIFT-based systems to verify the validity of the input data, so as to achieve high reliability.**

***Index Terms*—SIFT, keypoint removal, keypoint injection, convex optimization**

I. Introduction

In many pattern recognition and multimedia security systems, image local feature extraction and description become indispensable [2]–[9]. Due to its excellent robustness against partial occlusion, clutter, noise, lighting changes, and geometric transformation, *Scale Invariant Feature Transform* (SIFT) [10], as one of the methods for extracting image local features, has become extremely popular. SIFT has been extensively employed in many areas such as *Content Based Image Retrieval* (CBIR) systems [3], [11]–[13], 3D scene modeling [14], [15] and copy-move forgery detection applications [4], [8], [9], [16].

Essentially, the success of SIFT in these areas relies on the assumption that SIFT keypoints and the associated feature descriptors cannot be severely damaged without seriously distorting the image. Nevertheless, recent studies showed that advanced malicious attacks are capable of destroying image features in the scale space, including SIFT ones [17]–[25]. This could threaten the reliability and security of many systems built upon the SIFT features. For instance, a criminal may erase the SIFT keypoints of his ID photo, and then could successfully pass the security checking system designed over the SIFT feature domain [26], [27]. Another example is the copy-move forgery detection based on SIFT feature [7]–[9], [25]. If the SIFT keypoints in the cloned regions can be inhibited, then the forged images could escape from being detected, and will be treated as untouched ones.

The pioneer study on the security of SIFT was conducted by Hsu *et al.* [17], who attempted to inhibit a SIFT keypoint by duplicating another local extremum in the detection region. Later, Do *et al*. [28] argued that such attack was not enough to be a threat for a SIFT-based system, due to the *New Keypoint Generation* (NKG) problem; namely, new SIFT keypoints are produced in the neighborhood of the original ones, and they still can be matched with a high probability. To boost the SIFT keypoint removal performance, Do *et al.* proposed three methods. The first one called *Removal with Minimum local Distortion* (RMD) was designed to force the contrast value of each keypoint to be lower than the pre-defined contrast threshold [18]. However, this method tends to introduce severe artifacts into the resulting image, and the NKG problem still remains unsolved. The second strategy *Global Smoothing and Local Smoothing* (GSLS) tries to erase SIFT keypoints through globally and locally smoothing the images [18]. One of the disadvantages of this technique is that the resulting image is prone to be over-smoothed. The third one aims to change the orientation of each keypoint, making the modified SIFT descriptor difficult to be matched [20]. Lu and Hsu [21] more formally addressed the problem of SIFT keypoint removal by constructing a constrained optimization framework. As will become clear soon, the constraints incorporated into their optimization framework are too restricted, which seriously narrows the solution space, leading to large distortion. Recently,

This work was supported in part by the Macau Science and Technology Development Fund under grants FDCT/009/2013/A1, FDCT/046/2014/A1, FDCT/100/2012/A3, FDCT/026/2013/A, in part by the Research Committee at University of Macau under grants MRG007/ZJT/2015/FST, MRG021/ZJT/2013/FST, MYRG2014-00031-FST, MYRG205(Y1-L4)-FST11-TYY, MYRG187(Y1-L3)-FST11-TYY, MYRG2015-00049-FST, MYRG2015-00050-FST and RDG009/FST-TYY/2012, in part by the National Science Foundation of China under grants 61402547, 61300110 and 61273244, in part by the Macau-China join Project 008-2014-AMJ, and in part by the Fundamental Research Funds for the Central Universities under grant HIT.NSRIF.2015067.

Yuanman Li, Jiantao Zhou and Yuan Yan Tang are with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau. e-mails: {yb57410, jtzhou, yytang}@umac.mo

An Cheng is with Meitu, Inc. e-mail: ca@meitu.com

Xianming Liu is with the School of Computer Science and Technology, Harbin Institute of Technology. e-mail: xmliu.hit@gmail.com

The material in this paper was orally presented in part at the IEEE International Conference on Multimedia Expo (ICME), Torino, Italy, June 29 - July 3, 2015 [1]. (Corresponding author: Jiantao Zhou)

Amerini *et al.* [23] proposed the *CLassification-Based Attack* (CLBA), which first classifies the SIFT keypoints and then applies a different removal approach for each class.

It is observed that a standalone removal attack inevitably leaves cues in the resulting image, as few (or even no) SIFT keypoints exist in textured regions. Such abnormal phenomenon can be easily exposed to a well-designed forensic detector [29]. To solve this challenge, some works proposed to inject fake SIFT keypoints into the previously cleaned image, which has undergone removal operations [18], [21], [30]. Specifically, by adopting a strategy similar to their removal method RMD, Do *et al.* [18] suggested an injection method called *Forging new keypoint with Minimum local Distortion* (FMD) through a contrast enhancement mechanism. Amerini *et al.* later found that improved injection performance can be achieved by resorting to locally adaptive contrast enhancement techniques [30]. Further, Lu and Hsu [21] designed a coarse-to-fine descriptor searching strategy, based on which the descriptors of the injected keypoints are claimed to be able to match with those of a targeting image. Unfortunately, a recent study [29] demonstrated that most of the existing injection methods are incapable of fully concealing the SIFT removal footprints, which still can be revealed by a sophisticated forensic detector. Their proposed *Keypoint-to-Corner Ratio* (KCR) detector was shown to be very effective in discovering the SIFT keypoint removal traces, even if injection has been conducted. A key factor leading to the success of KCR detector is that the spatial distribution of the injected keypoints dramatically differs from that of the original ones.

In this paper, we study the SIFT keypoint removal problem by proposing a new constrained optimization framework, where we design a set of novel constraints to specifically address the local extrema suppression and the NKG problem simultaneously. We show that the ideal constraints are unfortunately non-convex. To make the computation feasible, we adopt a convex relaxation technique to convexify the original non-convex problem, while maximally preserving the solution space. We then address the problem of concealing the footprints of SIFT keypoint removal. We design a three-phase SIFT keypoint injection approach, which can re-introduce a large number of fake SIFT keypoints into the previously cleaned image, with spatial distribution similar to that of the original ones. Extensive experiment results are provided to show the superior performance of our proposed methods against the state-of-the-art techniques. Furthermore, we demonstrate that the combination of our proposed SIFT keypoint removal and injection approaches can successfully defeat the powerful KCR detector [29].

Difference from conference version: Portions of the work presented in this paper have previously appeared in [1] as a conference version. We have significantly revised and clarified the paper, and improved many technical details compared with [1]. The primary improvements can be summarized as follows. First of all, we provide a new Section IV to present a novel SIFT keypoint injection method via convex optimization. One of the design goals is to defeat the KCR detector [29]. Secondly, in Section V where we give the experimental results, we compare our proposed techniques with all the existing state-of-the-art methods, to more thoroughly demonstrate the superior performance. Thirdly, we evaluate different SIFT keypoint removal and injection approaches against the KCR detector. We show that the proposed SIFT keypoint removal and injection strategy can successfully escape from being detected by the KCR detector. Finally, we illustrate the usefulness of our method through a case study of image copy-move anti-forensics, in which we also offer the comparison with the state-of-the-art techniques.

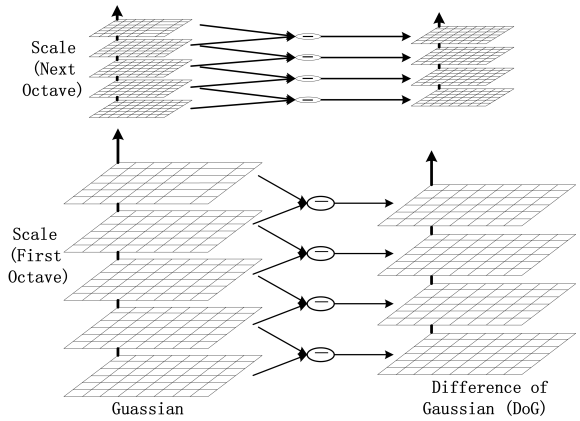


Fig. 1. The schematization of Gaussian-blurred images (left) and difference of Gaussians (DoG) images (right), where the convolved images are grouped by octave.

The rest of the paper is organized as follows. Section II briefly presents the SIFT algorithm. In Section III and IV, we describe the SIFT keypoint removal and injection frameworks, respectively. Section V gives extensive experimental results to validate the effectiveness of our proposed methods. We further demonstrate the superior performance of our techniques through a case study of image copy-move anti-forensics in Section VI, and finally we conclude in Section VII.

II. Preliminary of SIFT

SIFT is one of the most popular algorithms in computer vision to extract and describe image local features [10]. The SIFT algorithm can be divided into two stages: i) keypoint identification via extrema detection in the scale space, and ii) feature descriptor generation.

Fig. 1 depicts the construction of the scale space. For a specific octave v, the input image $\mathbf{I}_v$ is obtained through down-sampling or up-sampling the original image $\mathbf{I}$ by a factor ρ. Specifically,

$$\mathbf{I}_v(x,y) = \mathbf{I}(\lceil \rho \cdot x \rceil, \lceil \rho \cdot y \rceil);\ x \in \big[1, \frac{M}{\rho}\big] \cap \mathcal{Z}, y \in \big[1, \frac{N}{\rho}\big] \cap \mathcal{Z}$$

where M and N are the numbers of rows and columns of image $\mathbf{I}$, respectively. Then $\mathbf{I}_v$ is repeatedly convolved with Gaussian filters at multiple scales, to generate successive Gaussian-blurred images. The candidate keypoints are taken as the extreme points of the *Difference of Gaussians* (DoG) domain. More formally, for a fixed octave v, the DoG image of scale s is defined as

$$D_{\mathbf{I}}(x,y,\sigma_s) = L_{\mathbf{I}}(x,y,\sigma_{s+1}) - L_{\mathbf{I}}(x,y,\sigma_s) \tag{1}$$

where $L_{\mathbf{I}}(x, y, \sigma_s)$ represents the Gaussian-blurred image given by

$$L_{\mathbf{I}}(x, y, \sigma_s) = \mathbf{I}_v(x, y) \otimes G(x, y, \sigma_s) \tag{2}$$

and

$$G(x, y, \sigma_s) = \frac{1}{2\pi\sigma_s^2} e^{-(x^2+y^2)/2\sigma_s^2} \tag{3}$$

Letting $\mathbf{x} = (x, y, \sigma_s)$, we can write $D_{\mathbf{I}}(\mathbf{x}) \triangleq D_{\mathbf{I}}(x, y, \sigma_s)$. Each DoG value $D_{\mathbf{I}}(\mathbf{x})$ is compared with its 26 neighbors within a $3 \times 3 \times 3$ cube centered at itself. If $D_{\mathbf{I}}(\mathbf{x})$ is a local extremum (minimum or maximum), then $\mathbf{x}$ is selected as a candidate keypoint. All the candidate keypoints are further refined according to a contrast threshold and an edge threshold. At the stage ii), a 128-dimensional descriptor is calculated and assigned for each survived keypoint, encoding its surrounding information in the scale space. For more details about SIFT, please refer to [10].

III. SIFT KEYPOINT REMOVAL VIA CONVEX RELAXATION

In this section, we present our technique *Removal via Convex Relaxation* (RCR) to effectively remove the SIFT keypoints. Clearly, a well designed SIFT keypoint removal algorithm should simultaneously satisfy the following two design criteria: 1) the number of SIFT keypoints is significantly reduced, and 2) high quality of the resulting image is achieved. These two criteria are fundamentally conflicting with each other: more severe SIFT keypoint removal generally leads to larger distortion. To meet the first design criterion, we propose to suppress local extrema in the scale space, and avoid to generate new SIFT keypoints in a local cuboid.

As the SIFT algorithm is completely transparent even for a malicious attacker, the SIFT keypoints can be straightforwardly localized in the scale space. For a fixed octave, we denote $\mathbf{k}_o = (x_o, y_o, \sigma_{s_o})$ as the index of a generic SIFT keypoint in the scale space. Let

$$\mathcal{S}_o = \Big\{(x, y, \sigma_s)\Big| |x - x_o| \le 1, |y - y_o| \le 1, |s - s_o| \le 1, \\ x, y, s \in \mathcal{Z}\Big\} \tag{4}$$

be the index set of all the 27 points in the $3 \times 3 \times 3$ cube centered at $\mathbf{k}_o$. As $\mathbf{k}_o$ is a keypoint, it is either a maximum or a minimum in $\mathcal{S}_o$, implying that one of the following two inequalities must hold

$$D_{\mathbf{I}}(\mathbf{k}_o) > D_{\mathbf{I}}(\mathbf{x}), \forall \mathbf{x} \in \mathcal{S}_o \setminus \{\mathbf{k}_o\} \tag{5}$$

or

$$D_{\mathbf{I}}(\mathbf{k}_o) < D_{\mathbf{I}}(\mathbf{x}), \forall \mathbf{x} \in \mathcal{S}_o \setminus \{\mathbf{k}_o\} \tag{6}$$

In order to remove the keypoint $\mathbf{k}_o$, our strategy is to violate the above two inequalities defined over the DoG domain simultaneously. We should also bear in mind that the SIFT keypoint removal needs to be carried out with minimum effect on the image quality, which is measured over the pixel domain. To this end, we first extract a local image patch $\mathbf{p}_o$ of size $P \times P$ centered at (x_o, y_o) from the input image $\mathbf{I}$, where $P = 7$ in our implementation. Mathematically, we write $\mathbf{p}_o = \mathbf{E}_o \circ \mathbf{I}$, where $\mathbf{E}_o$ is the operator for extracting the image patch $\mathbf{p}_o$. Our target now is to produce a new image patch $\hat{\mathbf{p}}_o$ such that in the new image $\hat{\mathbf{I}}$ accommodating $\hat{\mathbf{p}}_o$, $\mathbf{k}_o$ is no longer a SIFT keypoint in the scale space, and no new SIFT keypoints are generated in its surroundings. Obviously, we have $\hat{\mathbf{p}}_o = \mathbf{E}_o \circ \hat{\mathbf{I}}$, and all the remaining pixels in $\hat{\mathbf{I}}$ are the same as the ones in the available $\mathbf{I}$. The SIFT keypoint removal can then be formulated into the following *generic* constrained optimization problem

$$\begin{aligned} &\min_{\hat{\mathbf{p}}_o} ||\mathbf{p}_o - \hat{\mathbf{p}}_o||_2^2 \\ &\text{s.t. } (C.1) : \mathbf{k}_o \text{ is not an extremum in } \mathcal{S}_o \text{ of } \hat{\mathbf{I}} \\ &\quad\ (C.2) : \text{no new keypoints generated} \end{aligned} \tag{7}$$

The objective function is straightforward, aiming at minimizing the ℓ_2 distortion between the resulting patch and the original one. We may also use some other distortion metrics, e.g., SSIM [31]. However, due to the simplicity and convexity, ℓ_2 distortion metric is the most natural choice. Our contribution in the proposed keypoint removal approach mainly lies in the determination of the two constraints $(C.1)$ and $(C.2)$ in an appropriate way. Clearly, to make the above optimization problem tractable, it is necessary to make both $(C.1)$ and $(C.2)$ convex, which permits efficient numerical implementations. It should be noted that the determination of both $(C.1)$ and $(C.2)$ is highly non-trivial, and appropriately determined constraints could lead to remarkably improved performance. This is similar to the image restoration problems, in which most of the methods are based on optimization; but different constraints result in significantly different restoration performance.

A. Determination of the condition $(C.1)$

The constraint $(C.1)$ imposed in (7) is to ensure that $\mathbf{k}_o$ is no longer an extremum (and hence a keypoint) within $\mathcal{S}_o$ in the new image $\hat{\mathbf{I}}$. To this end, we propose to violate the inequalities given in (5) and (6) simultaneously. Specifically, the constraint $(C.1)$ can be expressed as the following inequality

$$\alpha_o \le D_{\hat{\mathbf{I}}}(\mathbf{k}_o) \le \beta_o, \tag{8}$$

where

$$\alpha_o = \min_{\mathbf{x} \in \mathcal{S}_o \setminus \{\mathbf{k}_o\}} D_{\hat{\mathbf{I}}}(\mathbf{x}), \tag{9}$$

$$\beta_o = \max_{\mathbf{x} \in \mathcal{S}_o \setminus \{\mathbf{k}_o\}} D_{\hat{\mathbf{I}}}(\mathbf{x}) \tag{10}$$

The two inequalities in (8) hold because $\mathbf{k}_o$ is not identified as a keypoint if more than one extremum exists within the same $\mathcal{S}_o$. Unfortunately, as shown in Appendix, the constraint $(C.1)$ defined in (8) is non-convex, making the underlying optimization problem intractable. To resolve this challenge, we here resort to a convex relaxation technique to convexify the non-convex constraint in (8). Clearly, the relaxation should be conducted in a way to maximally preserve the solution space. This is because too strong relaxation, though capable of

obtaining convex constraint, causes the solution space far away from the truly optimal solution, leading to large distortion.

The difficulty of evaluating α_o and β_o in (9) and (10) arises from the fact that the minimization/maximization is taken over the unknown $\hat{\mathbf{I}}$. A viable solution to convexify the constraint in (8) is to approximate both α_o and β_o from the available $\mathbf{I}$. It is noticed that when $\hat{\mathbf{I}}$ and $\mathbf{I}$ are close to each other, the order in the scale space (relative relationship between one DoG value and its surroundings) tends to be preserved, though the DoG values may vary with large magnitudes. In fact, some recent studies showed that the order-preserving technique could be a powerful tool to regularize various optimization problems, e.g., in sparse coding [32], to achieve superior performance. Such order-preserving property motivates us to estimate the locations in $\mathcal{S}_o \setminus \{\mathbf{k}_o\}$ that correspond to the minimum and maximum of $D_{\hat{\mathbf{I}}}(\mathbf{x})$ from the available $\mathbf{I}$.

More specifically, we define

$$\mathbf{x}_{\min} = \arg \min_{\mathbf{x} \in \mathcal{S}_o \setminus \{\mathbf{k}_o\}} D_{\mathbf{I}}(\mathbf{x}), \tag{11}$$

where the minimization is conducted over the available $\mathbf{I}$, instead of over the unknown $\hat{\mathbf{I}}$. Similarly, we define

$$\mathbf{x}_{\max} = \arg \max_{\mathbf{x} \in \mathcal{S}_o \setminus \{\mathbf{k}_o\}} D_{\mathbf{I}}(\mathbf{x}) \tag{12}$$

Upon obtaining the locations $\mathbf{x}_{\min}$ and $\mathbf{x}_{\max}$, we now can estimate α_o and β_o according to the aforementioned order-preserving strategy through

$$\hat{\alpha}_o = D_{\hat{\mathbf{I}}}(\mathbf{x}_{\min}) \tag{13}$$

$$\hat{\beta}_o = D_{\hat{\mathbf{I}}}(\mathbf{x}_{\max}), \tag{14}$$

where the indexes $\mathbf{x}_{\min}$ and $\mathbf{x}_{\max}$ are calculated from $\mathbf{I}$. It should be pointed out that here the subscript of the DoG function is $\hat{\mathbf{I}}$, instead of $\mathbf{I}$. In other words, we only use the relative order information, while not the exact DoG values of $\mathbf{I}$.

Eventually, we can design the condition $(C.1)$ as a relaxed version of (8), which can be expressed as

$$\hat{\alpha}_o \le D_{\hat{\mathbf{I}}}(\mathbf{k}_o) \le \hat{\beta}_o \tag{15}$$

As $\mathbf{I}$ is available, both $\mathbf{x}_{\min}$ and $\mathbf{x}_{\max}$ in (13) and (14) are fixed. Considering the fact that the DoG function $D_{\hat{\mathbf{I}}}(\mathbf{x})$ is linear with respect to $\hat{\mathbf{I}}$, the above constraint $(C.1)$ is linear as well, and hence convex.

B. Determination of the condition $(C.2)$

Though we can eliminate the keypoint $\mathbf{k}_o$ by imposing the constraint $(C.1)$, we still cannot guarantee that new SIFT keypoints will not be generated in the surroundings. Such *New Keypoint Generation* (NKG) problem could severely degrade the SIFT keypoint removal performance, as the newly generated keypoints around the original one still have a high chance of being matched. Do *et. al* [18] showed that correct matches can be effectively destroyed if the keypoints are shifted in the scale space with large offsets. With this in mind, we design the constraint $(C.2)$ to make sure that no SIFT keypoints exist in a local cuboid $\mathcal{T}_o$ in the scale space. To be consistent with the setting of [10], we assume that there are 5 scales within each octave, indexed by $s = 0$ to $s = 4$, respectively. It is worthy to note that the SIFT keypoints can only be generated within the inner 3 scales, i.e., $s = 1, 2, 3$. We set the size of the local cuboid $\mathcal{T}_o$ to be $U \times U \times 3$, where $U = 7$ in our experiment. The schematization of $\mathcal{T}_o$ *constructed* by $\mathbf{k}_o$ is shown in Fig. 2. Formally, for a fixed octave, $\mathcal{T}_o$ is defined as

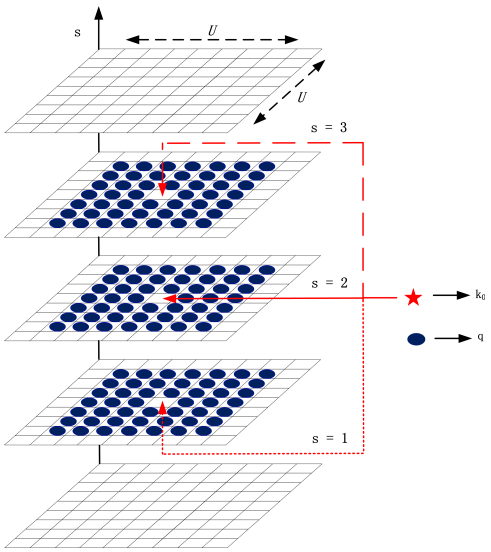


Fig. 2. The $U \times U \times 3$ sized cuboid *constructed* by $\mathbf{k}_o$.

$$\mathcal{T}_o = \left\{(x, y, \sigma_s) \middle| |x - x_o| \le \frac{U-1}{2}, |y - y_o| \le \frac{U-1}{2}, 1 \le s \le 3, x, y, s \in \mathcal{Z}\right\} \setminus \left\{\mathbf{k}_o\right\} \tag{16}$$

Note that there may exist some other SIFT keypoints in the cuboid $\mathcal{T}_o$, besides the newly generated ones. If this happens, we also remove them by imposing the following condition $(C.2)$. For each point $\mathbf{q} \in \mathcal{T}_o$, we consider the following two cases: 1) it is *not* an extremum in $\mathbf{I}$; and 2) it is an extremum in $\mathbf{I}$. We will address these two cases separately below.

For the Case 1), i.e., $\mathbf{q}$ is not an extremum in the $3 \times 3 \times 3$ cube $\mathcal{S}_q$ in the scale space centered at $\mathbf{q}$, where $\mathcal{S}_q$ can be similarly defined as (4), we compute

$$\mathbf{x}^q_{\min} = \arg \min_{\mathbf{x} \in \mathcal{S}_q} D_{\mathbf{I}}(\mathbf{x}), \tag{17}$$

$$\mathbf{x}^q_{\max} = \arg \max_{\mathbf{x} \in \mathcal{S}_q} D_{\mathbf{I}}(\mathbf{x}) \tag{18}$$

By using a similar strategy of deriving (15), the condition to ensure that $\mathbf{q}$ will not become a new keypoint can be written as

$$D_{\hat{\mathbf{I}}}(\mathbf{x}^q_{\min}) \le D_{\hat{\mathbf{I}}}(\mathbf{q}) \le D_{\hat{\mathbf{I}}}(\mathbf{x}^q_{\max}) \tag{19}$$

Obviously, this constraint is convex with respect to $\hat{\mathbf{I}}$.

For the Case 2), i.e., $\mathbf{q}$ itself is also an extremum in $\mathbf{I}$, we cannot directly apply (17) and (18) to calculate $\mathbf{x}^q_{\min}$ and $\mathbf{x}^q_{\max}$. Otherwise, (19) is always satisfied, because one of the inequalities must hold with equality. To tackle this issue, our strategy is to exclude $\mathbf{q}$ from $\mathcal{S}_q$ when calculating $\mathbf{x}^q_{\min}$ and $\mathbf{x}^q_{\max}$. Specifically, we have

$$\mathbf{x'}^q_{\min} = \arg \min_{\mathbf{x}\in\mathcal{S}_q\setminus\{\mathbf{q}\}} D_{\mathbf{I}}(\mathbf{x}), \tag{20}$$

$$\mathbf{x'}^q_{\max} = \arg \max_{\mathbf{x}\in\mathcal{S}_q\setminus\{\mathbf{q}\}} D_{\mathbf{I}}(\mathbf{x}) \tag{21}$$

Then, we can design the condition of making $\mathbf{q}$ no longer an extremum (hence a keypoint) as

$$D_{\hat{\mathbf{I}}}(\mathbf{x'}^q_{\min}) \le D_{\hat{\mathbf{I}}}(\mathbf{q}) \le D_{\hat{\mathbf{I}}}(\mathbf{x'}^q_{\max}) \tag{22}$$

Integrating both the constraints $(C.1)$ and $(C.2)$, we finally arrive at the convex optimization problem for removing the SIFT keypoint $\mathbf{k}_o$

$$\begin{aligned} &\min_{\hat{\mathbf{p}}_o} ||\mathbf{p}_o - \hat{\mathbf{p}}_o||_2^2 \qquad (23)\\ \text{s.t}\quad &\hat{\alpha}_o \le D_{\hat{\mathbf{I}}}(\mathbf{k}_o) \le \hat{\beta}_o\\ &D_{\hat{\mathbf{I}}}(\mathbf{x}^q_{\min}) \le D_{\hat{\mathbf{I}}}(\mathbf{q}) \le D_{\hat{\mathbf{I}}}(\mathbf{x}^q_{\max}), \forall \mathbf{q} \in \mathcal{T}_o \bigcap \mathcal{K}^c\\ &D_{\hat{\mathbf{I}}}(\mathbf{x'}^q_{\min}) \le D_{\hat{\mathbf{I}}}(\mathbf{q}) \le D_{\hat{\mathbf{I}}}(\mathbf{x'}^q_{\max}), \forall \mathbf{q} \in \mathcal{T}_o \bigcap \mathcal{K} \end{aligned}$$

where

$$\mathcal{K} = \Big\{(x, y, \sigma_s)\Big|(x, y, \sigma_s) \text{ is a keypoint of } \mathbf{I}\Big\}, \tag{24}$$

$\mathcal{K}^c$ is the complementary set of $\mathcal{K}$, and $\hat{\mathbf{p}}_o = \mathbf{E}_o \circ \hat{\mathbf{I}}$. In our implementation, we use the '`fmincon`' function provided by Matlab v.R2013b to solve the above optimization problem.

C. Iterative process of removing SIFT keypoints

In the above two subsections, we have presented our proposed SIFT keypoint removal algorithm, which is capable of removing one SIFT keypoint $\mathbf{k}_o$, and prohibiting the existence of any SIFT keypoints in a local cuboid constructed by $\mathbf{k}_o$. We can sequentially apply the removal operations described in (23) for all identified SIFT keypoints. However, upon the completion of one round of processing, there may still exist some SIFT keypoints that cannot be removed. This is because we can only guarantee that no keypoints exist in a local cuboid of size $U \times U \times 3$. While outside that cuboid, no guarantee can be offered. More importantly, in the case that multiple keypoints are tightly clustered, different patches $\hat{\mathbf{p}}_o$'s may be overlapped spatially. Such interference among different patches could also lead to some new SIFT keypoints.

To solve these two problems, we here adopt an iterative strategy. After each round r, the resulting image $\hat{\mathbf{I}}_r$ will serve as the input image in the next round. Such process will be iteratively performed for several rounds until the keypoint removal performance is satisfactory or the iteration number exceeds a threshold (50 in our implementation).

D. Remarks on the comparison with [21]

In this subsection, we briefly explain the differences between our proposed SIFT keypoint removal method and Lu's method [21], where a constrained optimization framework was also developed. Since the objective functions in both methods are straightforward ℓ_2 distortion terms, the primary differences lie in the design of the constraints. As explained previously and will be supported by our experimental results, different constraints can lead to significantly distinct SIFT keypoint removal performance.

Specifically, the constraint suggested in [21] to suppress the SIFT keypoint is to make the minimum point and the second minimum point equal to the average of their original values, and hence, generate two minima in the same $3 \times 3 \times 3$ cube. This condition can be treated as a special case of our constraint $(C.1)$ given in (15) when the left inequality is satisfied with equality to the average of the minimum and the second minimum. Clearly, the solution space of the framework in [21] is much restricted, which would lead to much more severe distortions. The experimental results to be given in Section V will verify our conclusion.

Furthermore, to prevent from generating new keypoints, the strategy in [21] is to force all the points in the $3 \times 3 \times 3$ cube, except $\mathbf{k}_o$, to be no smaller than $D_{\hat{\mathbf{I}}}(\mathbf{k}_o)$. Nevertheless, this cannot fully solve the NKG problem. In order to determine whether a point in the scale space is a keypoint or not, we should compare all the 27 points in the cube centered at itself, instead of by $\mathbf{k}_o$. As a comparison, when designing the proposed condition $(C.2)$, we can ensure that no SIFT keypoints exist in a $U \times U \times 3$ cuboid, where U is a parameter striking a balance between the removal performance and the incurred distortion.

IV. Three-phase SIFT keypoint injection

As demonstrated in [29], a single keypoint removal attack inevitably leaves cues in the resulting image, which can be easily revealed by well-designed detectors, e.g., the *Keypoint-to-Corner Ratio* (KCR) detector [29] [1]. To conceal the traces of performing SIFT keypoint removal, several algorithms including *Anisotropic Diffusion* (AD) [33], *Brightness Preserving Fuzzy Histogram Enhancement* (BPFHE) [34], *Contrast Limited Adaptive Histogram Equalization* (CLAHE) [35], *Gaussian smoothing* and *Forging new keypoint with Minimum local Distortion* (FMD) [18], were designed to inject fake SIFT keypoints into the cleaned image which has undergone removal operations. Unfortunately, all these injection schemes are reported to be unsuccessful in escaping from being detected by the KCR detector [29].

In this section, we propose a novel injection approach called *Three-Phase Keypoint Injection* (TPKI), which can inject a large number of fake SIFT keypoints with small incurred distortion on the resulting image. As will be clear shortly, our proposed TPKI can successfully defeat the KCR detector.

A. Design goals of the injection method

An appropriately designed SIFT keypoint injection method should satisfy the following requirements

[1]There are three detectors discussed in [29], among which KCR detector has been shown to be the most effective one according to a thorough comparison.

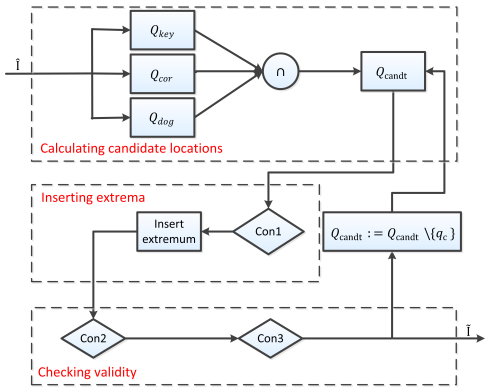


Fig. 3. The proposed injection framework. ***Con1***: check whether there exist any injected keypoints in the $3 \times 3 \times 3$ neighborhood centered at $\mathbf{q}_c$; ***Con2***: check whether $\mathbf{q}_c$ is survived as a final SIFT keypoint; ***Con3***: check whether the keypoint $\mathbf{q}_c$ introduces correct matches. If any one of these three conditions is not satisfied, injection at the current location is canceled. For brevity, we omit the 'false' branches of these three conditions.

i) The injected keypoints should not trigger new correct matches;
ii) The spatial distribution of the injected keypoints should be similar to that of the original ones;
iii) The incurred distortion should be minimized;
iv) A sufficient number of fake keypoints should be injected.

The first requirement is straightforward, as introducing new correct matches will degrade the performance of the previously applied SIFT keypoint removal. The second requirement is motivated by the observation that SIFT keypoints lie in proximity of corners. This property was extensively studied in [29] and serves as a key factor leading to the success of KCR detector. More specifically, KCR detector labels an image as forged if the ratio between the number of near-to-corner keypoints $N_{keypoints}$ and the number of corners $N_{corners}$ falls below a threshold T [29]

$$KCR = \log_{10}\left(\frac{N_{keypoints}}{N_{corners}}\right) \overset{?}{\leq} T \tag{25}$$

The third requirement ensures that the effect on the image quality by the keypoint injection is minimum. The fourth requirement is to eliminate the abnormal phenomenon that few (or even no) SIFT keypoints are located in textured regions. Another benefit of imposing this constraint is that injecting more fake keypoints can also increase the KCR value given in (25), potentially malfunctioning the KCR detector.

B. Details of our proposed injection framework

The systematic diagram of our proposed TPKI is depicted in Fig. 3. The input image is the previously cleaned image $\hat{\mathbf{I}}$, which has gone through the SIFT keypoint removal. The output image upon SIFT keypoint injection is denoted by $\tilde{\mathbf{I}}$. Our injection framework consists of three phases: calculating the candidate injection locations, inserting extrema, and checking validity, which will be detailed below.

1) Calculating the candidate injection locations: To fulfill the design goals described in Section IV-A, we construct a candidate injection location set $\mathcal{Q}_{candt}$. Only the locations belonging to $\mathcal{Q}_{candt}$ will be processed in the next two phases.

To prevent the injected keypoints from introducing new correct matches, as stated in the design requirement i), the injection location should be far away from the original keypoints. Let $\mathcal{Q}_{key}$ be the set containing all the locations with the distances to their nearest keypoints larger than d_1 ($d_1 = 8$ in our experiment). Specifically,

$$\mathcal{Q}_{key} = \left\{\mathbf{q} \middle| \ \forall \mathbf{x} \in \mathcal{K}, Dis(\mathbf{q}, \mathbf{x}) > d_1\right\} \tag{26}$$

where $\mathcal{K}$ defined in (24) is the set containing all the original keypoints, and $Dis(\cdot)$ computes the Euclidean distance of two spatial locations. Namely, only the first two dimensions of $\mathbf{q}$ and $\mathbf{x}$ enter in the distance computation.

In addition, to meet the design requirement ii), the spatial distribution of the injected keypoints should be similar to that of the original ones. Costanzo *et al.* [29] conducted extensive experiments, showing that the SIFT keypoints lie in proximity of corners for natural images. Letting $\mathcal{P}$ be the set recording all the corners in $\hat{\mathbf{I}}$, we define

$$\mathcal{Q}_{cor} = \left\{\mathbf{q} \middle| \ \exists \mathbf{x} \in \mathcal{P}, Dis(\mathbf{q}, \mathbf{x}) \leq d_2\right\} \tag{27}$$

which collects all the locations with the distances to their nearest corners smaller than d_2 ($d_2 = 3$ in our experiment).

Furthermore, to meet the design requirement iii), namely, minimize the incurred distortion, we impose additional constraints on the DoG magnitudes of the candidate injection locations. As stated in [10], the DoG magnitude of the SIFT keypoint cannot be smaller than the contrast threshold C, which is set to 4 in the well-known VLFEAT implementation [36]. Otherwise, it will be filtered out in the refinement stage. This implies that it is not desirable to inject SIFT keypoints to the locations with the DoG magnitudes much smaller than C. We hence only choose those locations with the DoG magnitudes bigger than $C - \tau$, where $\tau = 1$ in our implementation. Meanwhile, when deciding whether to inject a minimum or a maximum at a location $\mathbf{q}$, we should also bear in mind that the distortion should be minimized. Specifically, if $D_{\hat{\mathbf{I}}}(\mathbf{q}) - D_{\hat{\mathbf{I}}}(\hat{\mathbf{x}}^q_{min}) < \omega$ is satisfied, we insert a minimum, while if $D_{\hat{\mathbf{I}}}(\hat{\mathbf{x}}^q_{max}) - D_{\hat{\mathbf{I}}}(\mathbf{q}) < \omega$ holds, we insert a maximum. Here $\hat{\mathbf{x}}^q_{min}$ and $\hat{\mathbf{x}}^q_{max}$ can be similarly calculated as in (17) and (18) by replacing $\mathbf{I}$ with $\hat{\mathbf{I}}$, and $\omega = 0.5$ is a small constant. Combining all the conditions on the DoG magnitudes of the candidate locations, we define

$$\mathcal{Q}_{DoG} = \Big\{\mathbf{q} \Big| \ |D_{\hat{\mathbf{I}}}(\mathbf{q})| > C - \tau, \\ \min\left(D_{\hat{\mathbf{I}}}(\hat{\mathbf{x}}^q_{max}) - D_{\hat{\mathbf{I}}}(\mathbf{q}), D_{\hat{\mathbf{I}}}(\mathbf{q}) - D_{\hat{\mathbf{I}}}(\hat{\mathbf{x}}^q_{min})\right) < \omega\Big\} \tag{28}$$

To meet the above design requirements i)-iii) simultaneously, we intersect the three sets $\mathcal{Q}_{DoG}$, $\mathcal{Q}_{cor}$ and $\mathcal{Q}_{key}$, to form the set of candidate injection locations $\mathcal{Q}_{candt}$

$$\mathcal{Q}_{candt} = \mathcal{Q}_{DoG} \bigcap \mathcal{Q}_{cor} \bigcap \mathcal{Q}_{key} \tag{29}$$

Because the SIFT algorithm is completely transparent, the three sets $\mathcal{Q}_{DoG}$, $\mathcal{Q}_{cor}$ and $\mathcal{Q}_{key}$ can be readily calculated accompanying with the SIFT keypoint extraction from $\hat{\mathbf{I}}$. Note

that the candidate injection locations are calculated on all scales. We find that the size of $\mathcal{Q}_{candt}$ is generally more than 100 times larger than the number of original keypoints, providing us enough room for injecting fake keypoints. Hence, the design requirement iv) can be satisfied naturally.

2) Inserting extrema: Having $\mathcal{Q}_{candt}$ in hand, we now discuss how to make $\mathbf{q}_c = (x_c, y_c, \sigma_{s_c}) \in \mathcal{Q}_{candt}$ be an extremum in the $3 \times 3 \times 3$ cube with respect to the final resulting image $\tilde{\mathbf{I}}$. To this end, we first extract a local image patch $\mathbf{p}_c$ of size $P \times P$ ($P = 7$ in our experiments) centered at (x_c, y_c) from $\mathbf{I}$. Mathematically, $\mathbf{p}_c = \mathbf{E}_c \circ \mathbf{I}$, where $\mathbf{E}_c$ is the corresponding extraction matrix. Our target now is to generate a new image patch $\tilde{\mathbf{p}}_c$ such that, in $\tilde{\mathbf{I}}$ accommodating $\tilde{\mathbf{p}}_c$, $\mathbf{q}_c$ is an extremum. Clearly, we have $\tilde{\mathbf{p}}_c = \mathbf{E}_c \circ \tilde{\mathbf{I}}$, and all the remaining pixels in $\tilde{\mathbf{I}}$ are copied from the cleaned image $\hat{\mathbf{I}}$. Specifically, if the condition

$$D_{\hat{\mathbf{I}}}(\mathbf{x}_{max}^{q_c}) - D_{\hat{\mathbf{I}}}(\mathbf{q}_c) < D_{\hat{\mathbf{I}}}(\mathbf{q}_c) - D_{\hat{\mathbf{I}}}(\mathbf{x}_{min}^{q_c}) \tag{30}$$

is satisfied, we insert a maximum so as to minimize the distortion. Here $\mathbf{x}_{min}^{q_c}$ and $\mathbf{x}_{max}^{q_c}$ can be similarly calculated as (17) and (18) by replacing $\mathbf{I}$ with $\hat{\mathbf{I}}$, and $\mathcal{S}_q$ with $\mathcal{S}_{q_c}$, respectively.

The insertion of a maximum can be conducted by solving the following convex optimization problem

$$\begin{aligned} &\min_{\tilde{\mathbf{p}}_c} ||\mathbf{p}_o - \tilde{\mathbf{p}}_c||_2^2 \\ \text{s.t.} \quad &D_{\tilde{\mathbf{I}}}(\mathbf{q}) < D_{\tilde{\mathbf{I}}}(\mathbf{q}_c), \forall \mathbf{q} \in \mathcal{S}_{q_c} \setminus \{\mathbf{q}_c\} \\ &D_{\tilde{\mathbf{I}}}(\mathbf{q}_c) > C \end{aligned} \tag{31}$$

where $\mathcal{S}_{q_c}$ is defined in a similar way as in (4) by replacing $\mathbf{k}_o$ with $\mathbf{q}_c$, and C is the contrast threshold.

Clearly, the first constraint in (31) is to ensure that $\mathbf{q}_c$ is a maximum in $\mathcal{S}_{q_c}$. The second constraint is to guarantee that the DoG magnitude of $\mathbf{q}_c$ is bigger than C, so as to make the injected keypoint survive in the refinement stage. We should emphasize here that $\mathbf{q}_c$ is selected from $\mathcal{Q}_{candt}$.

Conversely, if the condition (30) does not hold, we insert a minimum instead by solving

$$\begin{aligned} &\min_{\tilde{\mathbf{p}}_c} ||\mathbf{p}_o - \tilde{\mathbf{p}}_c||_2^2 \\ \text{s.t.} \quad &D_{\tilde{\mathbf{I}}}(\mathbf{q}) > D_{\tilde{\mathbf{I}}}(\mathbf{q}_c), \forall \mathbf{q} \in \mathcal{S}_{q_c} \setminus \{\mathbf{q}_c\} \\ &D_{\tilde{\mathbf{I}}}(\mathbf{q}_c) < -C \end{aligned} \tag{32}$$

In addition, if $\mathbf{q}_c$ is successfully injected as a valid keypoint, then no extra extrema will be injected within its $3 \times 3 \times 3$ neighborhood. Note that there is at most one local maximum/minimum in any $3 \times 3 \times 3$ cube. If an injected keypoint exists in a $3 \times 3 \times 3$ cube and one forces to insert another local maximum/minimum, then the previously injected keypoint will disappear. Such double injection will introduce unnecessary distortion, and should be avoided. This is the reason for designing the "Con1" in Fig. 3.

3) Checking validity: It should be pointed out that the injection strategy through solving the aforementioned optimization problems (31) and (32) can only guarantee that $\mathbf{q}_c$ is a local extremum. This does not necessarily mean that $\mathbf{q}_c$ will eventually be a SIFT keypoint, as it still needs to go through two refinement stages. If $\mathbf{q}_c$ cannot pass, such injection will be given up. Another critical issue is that injecting keypoint to $\mathbf{q}_c$ may trigger correct matches with the original keypoints. If this happens, the current injection will also be canceled.

The injection process terminates when a sufficient number of fake keypoints are injected, or no extra candidate injection locations can be found. In our experiment, we check the number of injected keypoints upon the completion of each scale, for the sake of simplicity. Certainly, we can also check the condition after injecting each fake keypoint. Specifically, upon each injection, we should repetitively calculate the number of SIFT keypoints in the intermediate image, which is rather complicated. It is also worth noting that directly recording the number of injected keypoints does not work, as those injected fake keypoints may disappear due to the interference among different injections.

V. Experimental Results

In this section, we evaluate the proposed SIFT keypoint removal and injection methods. All the SIFT keypoints are extracted using the widely adopted SIFT-VLFEAT [36]. We set the associated peak and edge thresholds as 4 and 10, respectively, as adopted in [23], [29], [30], [37]. To be consistent with [23], [29], [30], only the keypoints in the first octave (i.e. octave 0) are considered in our experiments. The extension of our results to multiple octaves shall be straightforward.

A. Image data set

Our removal and injection experiments are conducted on two data sets. The first one consists of 8 standard test images: `Baboon`, `Barbara`, `Bridge`, `F16`, `Goldhill`, `Lena`, `Peppers`, and `Sailboat`, which are of size 512×512. This data set was also adopted in [21]. In the sequel, we refer this data set as `Dataset8`. The other one is `UCID-v2` corpus [38] consisting of 1338 uncompressed color images of size 512×384 with various characteristics. All the images in the two data sets are converted to gray-scale, prior to applying removal and injection attacks.

B. SIFT keypoint removal performance

We evaluate the performance of our proposed SIFT keypoint removal algorithm in terms of *Keypoint Remove Rate-Distortion* (KRR-D) metric. More specifically, the KRR is defined as

$$KRR = 1 - \frac{\#\text{ correctly matched keypoints after removal}}{\#\text{ number of original keypoints}} \tag{33}$$

The distortion is measured between the original image $\mathbf{I}$ and the one after keypoint removal $\hat{\mathbf{I}}$ in full-frame, in terms of the *Peak Signal-to-Noise Ratio* (PSNR) criterion.

We first perform the comparison among our proposed RCR, GSLS7 [18], RMD [18], CLBA [23] and Lu's method [21] on *Dataset8*. As the source code of [21] is not available, the results represented by the dotted lines ($*$) in Fig. 4 are extracted from Table 1 of [21]. In addition, for the sake of fairness, all the parameters in GSLS7, RMD and CLBA are carefully tuned to achieve the best KRR-D performance. As can be observed, our proposed RCR significantly outperforms

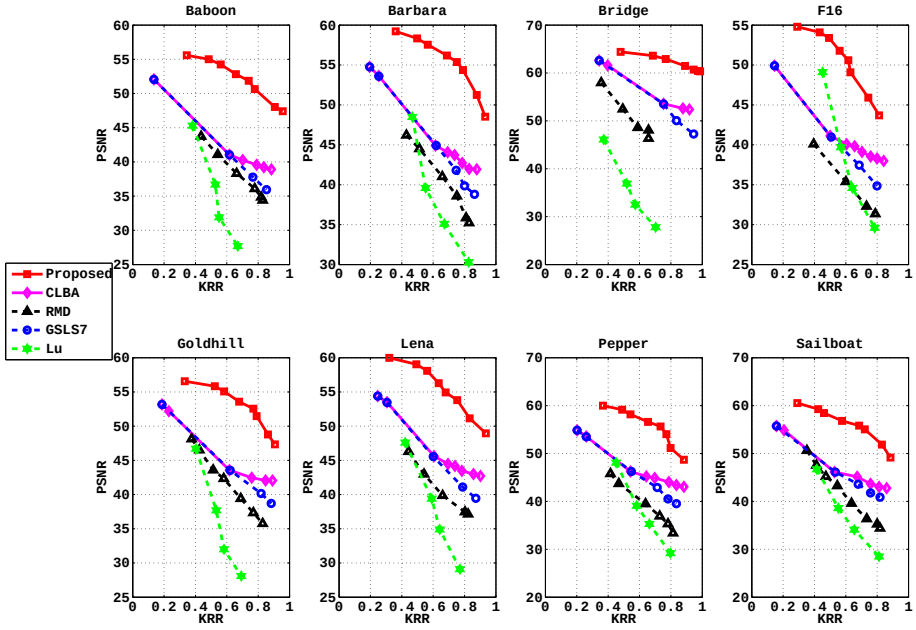
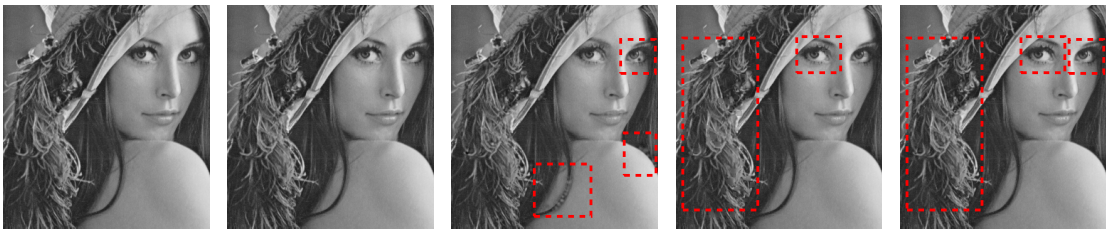
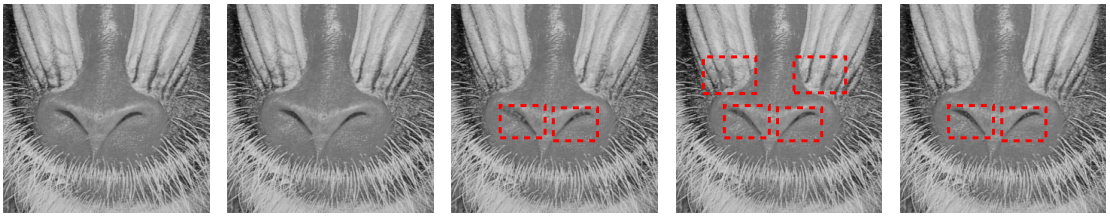


Fig. 4. Comparison with the proposed RCR, RMD [18], GSLS7 [18], CLBA [23] and Lu's method [21] in terms of KRR-D performance for 8 test images.



(a) Original (b) Proposed RCR (c) RMD [18] (d) GSLS7 [18] (e) CLBA [23]

Fig. 5. Visual quality comparison with proposed RCR, RMD [18], GSLS7 [18] and CLBA [23]. (a) original, (b) RCR (KRR 93.60%, PSNR 48.95 dB, SSIM 0.9990), (c) RMD (KRR 86.21%, PSNR 35.98 dB, SSIM 0.9876), (d) GSLS7 (KRR 87.19%, PSNR 39.45 dB, SSIM 0.9934), and (e) CLBA (KRR 89.66%, PSNR 42.47 dB, SSIM 0.9965)



(a) Original (b) Proposed RCR (c) RMD [18] (d) GSLS7 [18] (e) CLBA [23]

Fig. 6. Visual quality comparison with proposed RCR, RMD [18], GSLS7 [18] and CLBA [23]. (a) original, (b) RCR (KRR 95.66%, PSNR 47.40 dB, SSIM 0.9989), (c) RMD (KRR 82.89%, PSNR 34.40 dB, SSIM 0.9891), (d) GSLS7 (KRR 85.30%, PSNR 35.94 dB, SSIM 0.9875), and (e) CLBA (KRR 90.12%, PSNR 38.63 dB, SSIM 0.9931)

the other four competing methods in terms of KRR-D metric. Take Lena as an example. When KRR = 0.8, the PSNR gains of RCR over RMD, GSLS7 and CLBA are about 14 dB, 11 dB and 8 dB, respectively. Under the same KRR, the PSNR gain against Lu's method [21] reaches 23 dB, which is quite remarkable. This explains the importance of appropriately determining the constraints in the optimization framework, which is one of our major contributions in the proposed keypoint removal algorithm.

In addition to the KRR-D curves, we present the visual quality comparison in Figs. 5-6 among our proposed RCR, RMD, GSLS7 and CLBA, where full-frame PSNR and SSIM results are also provided. In the relatively smooth areas, all these four competing methods achieve pretty good results, because not many SIFT keypoints exist in these regions. However, for the textured areas, RCR obtains more visually

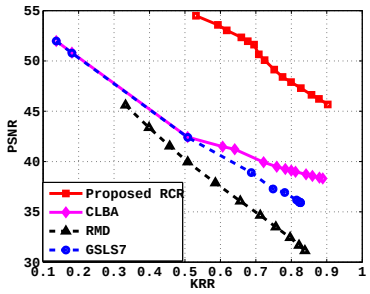


Fig. 7. Comparison with the proposed RCR, RMD [18], GSLS7 [18] and CLBA [23] in terms of average KRR-D performance over UCID-v2.

pleasing results than the other three competitors. The readers are invited to examine the highlighted regions enclosed in the red boxes of Fig. 5(c) and 6(c), where RMD tends to introduce severe artifacts around the highly textured regions. From Fig. 5(d) and Fig. 5(e), we can notice that most parts of the hair are highly smoothed, as GSLS7 and CLBA are prone to over-smoothing the fine details.

We also compare the SIFT keypoint removal performance of the proposed RCR, RMD, GSLS7 and CLBA for all the 1338 images in the UCID-v2. Fig. 7 reports the average KRR-D performance of these four methods. Again, the proposed RCR outperforms the other three methods by a big margin. For example, when KRR = 0.8, the PSNR gains of RCR over RMD, GSLS7 and CLBA are about 15 dB, 11 dB and 9 dB, respectively, which is quite significant.

C. SIFT keypoints injection performance

We now investigate the effectiveness of the proposed injection algorithm TPKI. We quantify the performance of injection algorithms using the following two metrics.

The first one is the *Keypoint Injection Rate-Distortion (KIR-D)*, where the KIR given below is defined in [37].

$$KIR = \frac{\#\text{New keypoints following injection}}{\#\text{keypoints before removal}} \times 100 \quad (34)$$

The distortion is measured in ℓ_2 sense, between the original image $\mathbf{I}$ and the one after keypoint injection $\tilde{\mathbf{I}}$ in full-frame. Obviously, an effective injection method should be capable of retaining a high KIR value while keeping the distortion small.

The second important metric is denoted by #Match, representing the number of keypoints in the resulting image which are correctly matched with the original ones. As can be expected, #Match should be as small as possible in accordance with the design requirement i) described in Section IV-A.

Our experiments of injection are conducted on both Dataset8 and UCID-v2. All the images are first attacked with our proposed removal technique RCR. The resulting images are then attacked with the proposed injection method TPKI. We call this combined attack strategy RCR+TPKI. We now compare the performance of RCR+TPKI with the existing GSLS7+FMD [18] and CLBA+FMD [37] in Table I. Here, #KP in the second column represents the numbers of keypoints in the original images, while all the other #KP's are the numbers of keypoints after applying the corresponding combined attacks. Both PSNR and SSIM results are provided.

It can be observed that RCR+TPKI significantly outperforms GSLS7+FMD and CLBA+FMD under both KIR-D and #Match criteria. Taking Baboon as an example, the KIR value achieved by our RCR+TPKI is around 5 times higher than that given by GSLS7+FMD and CLBA+FMD. Meanwhile, the PSNR gains over these two competitors are over 7 dB and 5 dB, respectively. We also notice that RCR+TPKI maintains much lowered #Match. In the last row of Table I, we report the average performance comparison over the images from UCID-v2. Similar conclusion can be drawn as that for Dataset8.

The visual quality comparison among the aforementioned three combined attacks is given in Fig. 8. As can be seen and also reported in [29], GSLS7+FMD and CLBA+FMD tend to introduce severe artifacts in the resulting images. In contrast, the visual distortion incurred by our RCR+TPKI is controlled to an imperceptible level, even though larger number of fake keypoints are injected.

One may also be interested in the behavior of some other combined attacks, e.g., GSLS7+TPKI, CLBA+TPKI and RCR+FMD. As the proposed injection algorithm TPKI is independent of removal methods, the combined strategies GSLS7+TPKI and CLBA+TPKI are also capable of injecting a sufficient number fake keypoints; however, the incurred distortions on the resulting images are larger compared with our proposed RCR+TPKI. For instance, the average performance of GSLS7+TPKI/CLBA+TPKI on UCID-v2 is: #KP = 347/361, #Match = 91/48, KIR = 74/80 and PSNR = 35.89/37.52. As can be seen from Table I, such performance is still inferior than that of RCR+TPKI. For another combination RCR+FMD, the average performance on UCID-v2 is: #KP = 129, #Match = 55, KIR = 21 and PSNR = 41.25, indicating that the number of injected keypoints is much lowered.

D. Performance against the KCR detector

To further evaluate the performance of different SIFT keypoint removal and injection attacks, in this subsection, we compare their capabilities against the KCR detector. Specifically, the following 7 types of attacks are evaluated

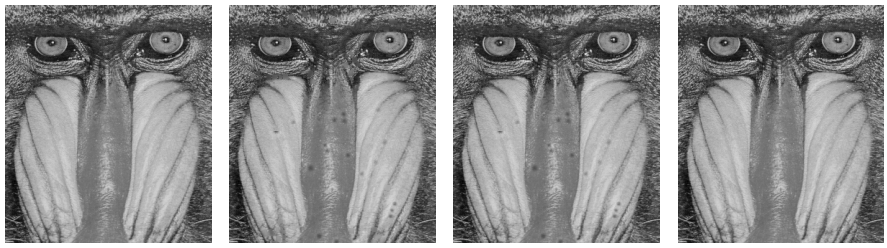
- RCR(60%\80%): remove 60%\80% of the keypoints using the proposed removal method RCR.
- RMD(60%\80%): remove 60%\80% of the keypoints using RMD [18].
- GSLS7(60%\80%): remove 60%\80% of the keypoints with GSLS7 [18].
- CLBA(60%\80%): remove 60%\80% of the keypoints using CLBA [37].
- RCR+TPKI: remove the keypoints to the greatest extent using RCR, and then inject fake keypoints using TPKI.
- GSLS7+FMD: remove the keypoints to the greatest extent using GSLS7, and then inject fake keypoints using FMD.
- CLBA+FMD: remove the keypoints to the greatest extent using CLBA, and then inject fake keypoints using FMD.

As in [29], the threshold T in KCR detector is set to be -1.9 to distinguish a forged image from the original,

TABLE I
PERFORMANCE OF INJECTION METHODS

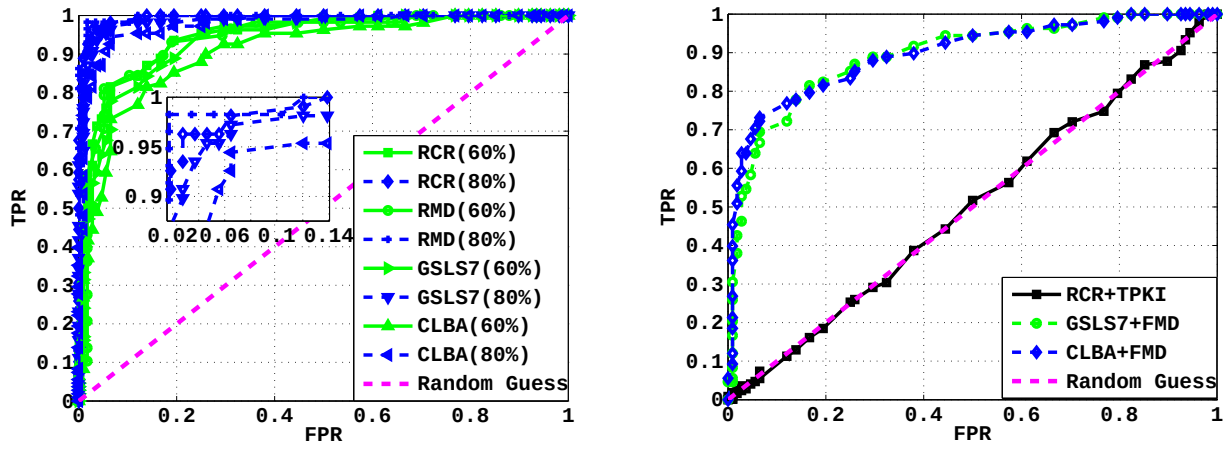
Image	#KP	GSLS7+FMD				CLBA+FMD				Proposed RCR+TPKI			
		#KP	#Match	KIR	PSNR (SSIM)	#KP	#Match	KIR	PSNR (SSIM)	#KP	#Match	KIR	PSNR (SSIM)
Baboon	415	175	60	20	35.41 (0.9836)	159	31	21	37.72 (0.9892)	469	13	103	43.26 (0.9980)
Barbara	205	148	42	40	38.03 (0.9883)	124	15	41	38.83 (0.9896)	239	10	108	44.46 (0.9973)
Bridge	73	93	6	115	40.10 (0.9870)	100	7	121	40.57 (0.9878)	125	1	164	52.30 (0.9989)
F16	448	171	56	19	32.55 (0.9680)	200	56	18	35.64 (0.9780)	510	44	91	39.20 (0.9912)
Goldhill	386	232	50	40	36.99 (0.9809)	239	34	43	38.78 (0.9847)	428	21	99	41.59 (0.9928)
Lena	203	139	21	44	37.32 (0.9831)	131	17	43	38.93 (0.9860)	267	15	124	44.03 (0.9964)
Peppers	188	141	35	55	35.73 (0.9771)	146	24	57	36.83 (0.9798)	262	14	126	44.57 (0.9966)
Sailboat	165	97	33	38	38.11 (0.9867)	92	20	39	38.83 (0.9881)	239	12	130	45.76 (0.9979)
UCID	354	170	93	17	35.17 (0.9821)	144	42	18	36.70 (0.9859)	355	28	85	40.43 (0.9940)

(a) Original (b) GSLS7+FMD [18] (c) CLBA+FMD [37] (d) RCR+TPKI



(e) Original (f) GSLS7+FMD [18] (g) CLBA+FMD [37] (h) RCR+TPKI

Fig. 8. Visual quality comparison of different combined attacks. (a) original Lena, (b) GSLS7+FMD (KIR 44%, PSNR 37.32 dB, SSIM 0.9831, #Match 21), (c) CLBA+FMD (KIR 43%, PSNR 38.93 dB, SSIM 0.9860, #Match 17), (d) RCR+TPKI (KIR 124%, PSNR 44.03 dB, SSIM 0.9964, #Match 15), (e) original Baboon, (f) GSLS7+FMD (KIR 20%, PSNR 35.41 dB, SSIM 0.9836, #Match 60), (g) CLBA+FMD (KIR 21%, PSNR 37.72 dB, SSIM 0.9892, #Match 31), (h) RCR+TPKI (KIR 103%, PSNR 43.26 dB, SSIM 0.9980, #Match 13).



(a) (b)

Fig. 9. ROC curves of different attacks. The enlarged version of the upper-left corner is also shown in (a) for better illustration.

untouched one. Under this configuration, all the images in UCID-v2 upon applying RCR+TPKI can avoid being detected by the KCR detector with 100% success rate. To gain deeper understanding, we draw the ROC curves for the above 7 types of attacks in Fig. 9. The threshold T varies in the interval $[-5, 0]$ with step size being 0.02. We use the

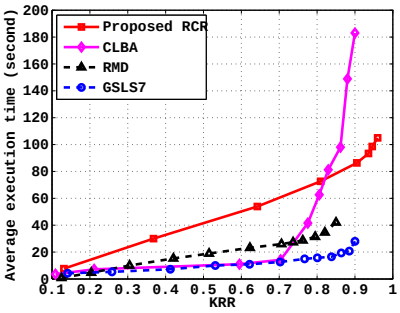


Fig. 10. The average execution time over Dataset8. For CLBA, the complexity of constructing the big patch database is not included.

original images as negative instances, and the forged images (attacked images) as positive instances. Fig. 9(a) shows the results for single removal attacks against the KCR detector. It can be seen that all these single removal attacks can be effectively detected. Further, as the keypoint removal rate increases, it becomes easier to detect the forged images. In Fig. 9(b), we give the results of the combined attacks against KCR detector. It is observed that, even incorporated with the injection approach FMD, GSLS7+FMD and CLBA+FMD still fail to be undetectable by KCR detector. For instance, when fixing the *False Positive Rate* (FPR) to be 0.1, the *True Positive Rate* (TPR)'s of these two combined attack strategies reach 0.70 and 0.78, respectively. There are two major reasons why these two attacks can be detected by the KCR detector: 1) the number of injected keypoints by FMD is generally much smaller than that of the original image; and 2) FMD takes no consideration about the spatial distribution of the injected fake keypoints. In contrast, the ROC curve of our proposed RCR+TPKI is very close to that of the random guess. This implies that the images attacked by RCR+TPKI succeed in defeating the KCR detector.

E. Computational complexity

We conclude this section by a brief discussion of the computational complexity of different SIFT keypoint removal algorithms. Fig. 10 reports the average execution time over Dataset8, where all the experiments are conducted on a desktop equipped with Core-i7 and 8-GB RAM. We can observe that GSLS7 and RMD are very fast, as smoothing operations and evaluating closed-form solutions can be efficiently implemented. For CLBA, the complexity is quite close to that of GSLS7 when KRR is below 0.7, because CLBA adopts GSLS7 for the first 10 iterations. As KRR becomes larger, the complexity of CLBA increases very quickly. This is due to the fact that CLBA includes the *collage* attack for the additional iterations, which involves patch searching over a big database consisting of 120,000 patches [23]. Since our proposed RCR needs to solve a set of constrained optimization problems, the computational complexity is higher than that of GSLS7 and RMD for all KRRs. Compared with CLBA, RCR is more time-consuming when KRR is less than 0.82. However, as KRR is further increased, CLBA becomes less efficient. Noticing the fact that SIFT keypoint removal is typically conducted offline (similar to the case of making forged images), the complexity may not be a crucial issue in many scenarios.

VI. Application to Copy-Move Anti-Forensics

In this section, we further show the superior performance of our proposed RCR+TPKI through a case study of anti-forensics of SIFT-based copy-move detection. Copy-move forgery is a popular means of making doctored images by cloning an area of an image onto another zone, probably accompanying with some appropriate geometric transformations [39], [40]. The state-of-the-art image copy-move forgery detector [9] has been proven to be quite effective in robustly detecting cloned regions via SIFT matching[2]. Conversely, the image copy-move anti-forensic techniques aim to dis-link the SIFT-matched keypoints from the cloned region to its source, while still maintaining high quality of the resulting image. As will be clear shortly, upon applying our attack strategy RCR+TPKI, the doctored image created by copy-move forgery will not be detected by the forensic detector in [9], and the resulting image is still of high quality.

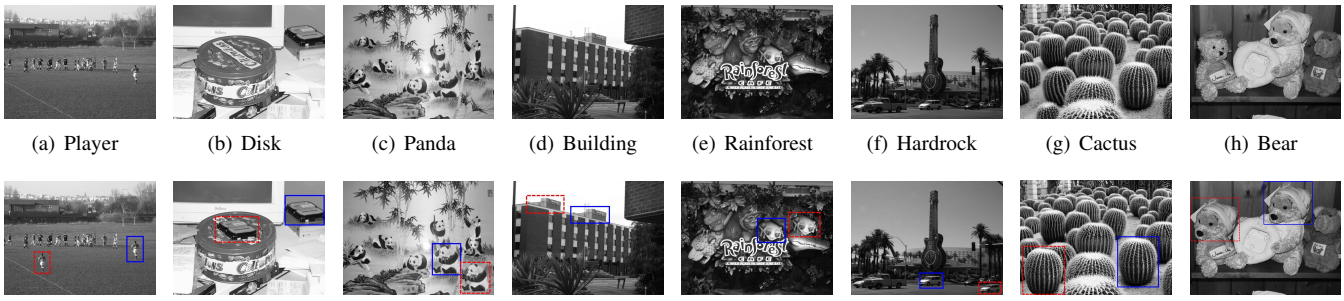
Specifically, given one forged image, we first apply the SIFT keypoint removal in the cloned region, and then inject fake keypoints inside. The resulting image is further checked with the copy-move forgery detector in [9] to see whether it can be detected or not. In Fig. 11, we illustrate 8 original images, and their forged versions through copy-move attacks. The performance of different combined attacks performed on the forged images is compared in Table II. It can be seen that the performance of the proposed RCR+TPKI significantly outperforms the other two combined attacks. Here, LPSNR and LSSIM represent local PSNR and local SSIM, respectively. They are calculated from the processed regions only, rather than from the whole image. This is because in the copy-move anti-forensics applications, we only need to remove and inject SIFT keypoints for the cloned regions.

In addition, we demonstrate the fine-grained analysis of a representative image in Fig. 12. As can be seen from Figs. 12(a), (b), the forged image is produced by copying one cactus and pasting it to another location. This kind of forgery can be effectively detected by the copy-move forgery detector [9], as demonstrated in Fig. 12(c), where many matching pairs exist between the cloned region and its original. Upon applying the proposed RCR+TPKI, such matching pairs disappear, and the injected fake SIFT keypoints still follow a similar spatial distribution as that of the original ones, as illustrated in Figs. 12(d), (e). We also show the results of GSLS7, GSLS7+FMD, CLBA and CLBA+FMD in Figs. 12 (f)-(i) for comparison purposes.

VII. Conclusion

In this work, we have investigated the security of SIFT against malicious attacks. We have designed an effective

[2]In addition to SIFT-based approaches, there are some other image copy-move forgery detection methods through dense sampling [41]–[45]. It was reported that some of them are able to achieve even better detection performance at the cost of higher complexity. The discussion of these image copy-move forgery detection schemes is beyond the scope of this work.



(a) Player (b) Disk (c) Panda (d) Building (e) Rainforest (f) Hardrock (g) Cactus (h) Bear

Fig. 11. The original images are shown in the first row, and the corresponding copy-move attacked images are given in the second row. The clone regions and the source regions are enclosed in red and blue boxes, respectively.

TABLE II
PERFORMANCE OF COMBINED ATTACK STRATEGIES FOR COPY-MOVE ANTI-FORENSICS. HERE, LPSNR AND LSSIM DENOTE LOCAL PSNR AND LOCAL SSIM CALCULATED FROM THE PROCESSED REGIONS, RESPECTIVELY.

Image	#KP	#Match	GSLS7+FMD			CLBA+FMD			Proposed RCR+TPKI		
			#KP	#Match	LPSNR(LSSIM)	#KP	#Match	LPSNR(LSSIM)	#KP	#Match	LPSNR(LSSIM)
Player	7	6	1	0	33.41 (0.9836)	2	0	30.17 (0.9810)	5	0	37.96 (0.9817)
Disk	50	6	45	0	45.64 (0.9960)	45	0	45.64 (0.9960)	52	0	49.89 (0.9986)
Panda	35	16	23	0	34.04 (0.9770)	25	0	32.77 (0.9789)	37	0	44.26 (0.9973)
Building	16	3	13	0	46.02 (0.9976)	13	0	46.02 (0.9976)	19	0	50.61 (0.9991)
Rainforest	29	10	20	0	35.32 (0.9869)	20	0	38.36 (0.9927)	33	0	46.15 (0.9976)
Hardrock	10	5	6	0	43.45 (0.9953)	6	0	40.92 (0.9927)	11	0	48.42 (0.9980)
Cactus	148	91	100	0	30.78 (0.9805)	93	0	31.35 (0.9849)	149	0	37.09 (0.9947)
Bear	127	93	89	0	36.38 (0.9789)	91	1	37.02 (0.9805)	138	0	43.46 (0.9951)

strategy to remove the SIFT keypoints while still maintaining high quality of the resulting image. This has been cast as a constrained optimization problem, where the constraints are well-designed to suppress the existence of local extrema and prevent generating new keypoints in a local cuboid in the scale space. Unfortunately, the optimization problem in the ideal case has been shown to be non-convex. To tackle such challenge, we have proposed a convex relaxation technique to approximate the original problem. Further, to eliminate the abnormal phenomenon that few (or even no) keypoints exist in textured regions, a novel keypoint injection technique TPKI has been proposed. It has been shown that TPKI is capable of injecting a large number of fake keypoints into the cleaned images at the cost of small distortion. Extensive experiments have been conducted to demonstrate that our proposed removal/injection method can achieve much better KRR-D/KIR-D performance than the state-of-the-art techniques. In addition, our proposed combined attack strategy RCR+TPKI has been proved to be able to defeat the KCR detector.

An implication of our results is that we cannot fully trust the images input to the SIFT-based system, especially in security application scenarios. Appropriate authentication needs to be performed to validate the data, before they are fed into the SIFT module. Otherwise, the decision made upon the extracted SIFT features could be groundless.

VIII. ACKNOWLEDGEMENT

We would like to thank Dr. Do from Singapore University of Technology and Design (SUTD) for sharing us the source codes of FMD, RMD and GSLS7. The authors also thank the Associate Editor Prof. Anderson Rocha and four anonymous reviewers for their helpful comments.

APPENDIX
PROOF SKETCH OF THE NON-CONVEXITY OF (8)

Let us first define four sets

$$\begin{aligned}\mathcal{P}_1 &= \left\{\hat{\mathbf{p}}_o \in \mathcal{R}^{P\times P} |\ D_{\hat{\mathbf{I}}}(\mathbf{k}_o) > \beta_o\right\}\\ \mathcal{P}_2 &= \left\{\hat{\mathbf{p}}_o \in \mathcal{R}^{P\times P} |\ D_{\hat{\mathbf{I}}}(\mathbf{k}_o) < \alpha_o\right\}\\ \mathcal{P}_3 &= \left\{\hat{\mathbf{p}}_o \in \mathcal{R}^{P\times P} |\ \alpha_o \le D_{\hat{\mathbf{I}}}(\mathbf{k}_o) \le \beta_o\right\}\\ \mathcal{P}_4 &= \left\{\hat{\mathbf{p}}_o \in \mathcal{R}^{P\times P}\right\}\end{aligned} \tag{35}$$

where α_o and β_o are given in (9) and (10), respectively, and $P = 7$ in our implementation.

It can be easily seen that

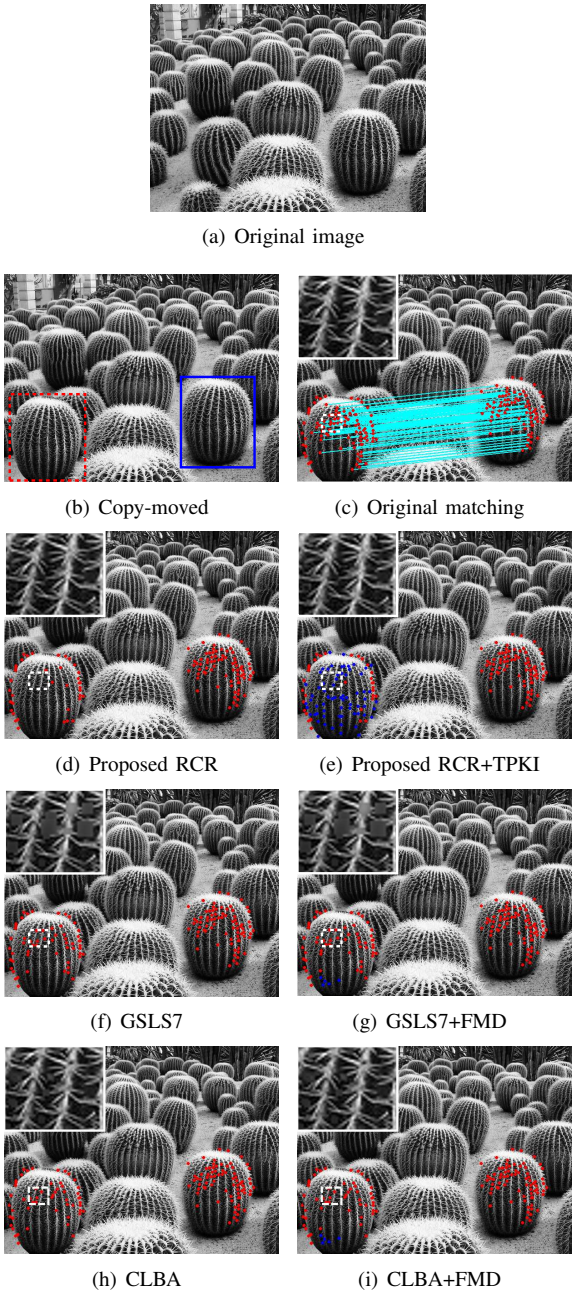
$$\mathcal{P}_4 = \mathcal{P}_1 \bigcup \mathcal{P}_2 \bigcup \mathcal{P}_3 \tag{36}$$

Clearly, the DoG function $D_{\hat{\mathbf{I}}}(\mathbf{x})$ is an affine and hence convex with respect to $\hat{\mathbf{p}}_o$. Due to the fact that maximization preserves convexity [46], $\beta_o = \max_{\mathbf{x}\in\mathcal{S}_o\setminus\{\mathbf{k}_o\}} D_{\hat{\mathbf{I}}}(\mathbf{x})$ is also convex with respect to $\hat{\mathbf{p}}_o$. Noticing the fact that $D_{\hat{\mathbf{I}}}(\mathbf{k}_o)$ is affine, we conclude that $\beta_o - D_{\hat{\mathbf{I}}}(\mathbf{k}_o)$ is convex; but certainly it is not affine. This implies that the set $\mathcal{P}_1$ is convex but not a halfspace [46]. Similarly, we can prove that $\mathcal{P}_2$ is convex but not a halfspace. As $\mathcal{P}_3 = \mathcal{P}_4 \setminus \{\mathcal{P}_1 \bigcup \mathcal{P}_2\}$, $\mathcal{P}_1$ and $\mathcal{P}_2$ are non-overlapping, and both $\mathcal{P}_1$ and $\mathcal{P}_2$ are convex but not halfspaces, we can conclude that $\mathcal{P}_3$ must be non-convex. This completes the proof.

REFERENCES

[1] A. Cheng, Y. Li, and J. Zhou, "SIFT keypoint removal via convex relaxation," in *Proc. IEEE Int. Conf. on Multimedia and Expo*. IEEE, 2015, pp. 1–6.

[2] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. on Computer Vision*, vol. 2, 1999, pp. 1150–1157.



(a) Original image

(b) Copy-moved (c) Original matching

(d) Proposed RCR (e) Proposed RCR+TPKI

(f) GSLS7 (g) GSLS7+FMD

(h) CLBA (i) CLBA+FMD

Fig. 12. Example of anti-forensics of copy-move forgery detection. The original SIFT keypoints and the injected ones are denoted by red and blue dots, respectively.

[3] B. J. H. Lejsek, F.H. Asmundsson and L. Amsaleg, "NV-tree: An efficient disk-based index for approximate search in very large high-dimensional collections," *IEEE Trans. on Pattern Anal. and Mach. Intell.*, vol. 31, no. 5, pp. 869–883, May 2009.

[4] J. Li, X. Li, B. Yang, and X. Sun, "Segmentation-based image copy-move forgery detection scheme," *IEEE Trans. on Inf. Forensics and Security*, vol. 10, no. 3, pp. 507–518, Mar. 2015.

[5] Z. Qin, J. Yan, K. Ren, C. W. Chen, and C. Wang, "Towards efficient privacy-preserving image feature extraction in cloud computing," in *Proc. ACM Int. Conf. on Multimedia*, 2014, pp. 497–506.

[6] C. Strecha, A. Bronstein, M. Bronstein, and P. Fua, "LDAHash: Improved matching with smaller descriptors," *IEEE Trans. on Pattern Anal. and Mach. Intell.*, vol. 34, no. 1, pp. 66–78, Jan. 2012.

[7] H. Huang, W. Guo, and Y. Zhang, "Detection of copy-move forgery in digital images using SIFT algorithm," in *Proc. IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, vol. 2, Dec. 2008, pp. 272–276.

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

[9] 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 Processing: Image Communication*, vol. 28, no. 6, pp. 659 – 669, 2013.

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

[11] J. J. Foo, J. Zobel, R. Sinha, and S. M. M. Tahaghoghi, "Detection of near-duplicate images for web search," in *Proc. ACM Int. Conf. on Image and Video Retrieval*, 2007, pp. 557–564.

[12] K. Grauman and T. Darrell, "Efficient image matching with distributions of local invariant features," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2005, pp. 627–634.

[13] Y. Ke, R. Sukthankar, and L. Huston, "Efficient near-duplicate detection and sub-image retrieval," in *Proc. ACM Int. Conf. on Multimedia*, vol. 4, no. 1, 2004, pp. 869–876.

[14] I. Skrypnyk and D. Lowe, "Scene modelling, recognition and tracking with invariant image features," in *Proc. IEEE and ACM Int. Symp. Mixed and Augmented Reality (ISMAR)*, Nov 2004, pp. 110–119.

[15] I. Gordon and D. Lowe, "What and where: 3D object recognition with accurate pose," in *Toward Category-Level Object Recognition*. Springer, 2006, vol. 4170, pp. 67–82.

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

[17] C.-Y. Hsu, C.-S. Lu, and S.-C. Pei, "Secure and robust SIFT," in *Proc. ACM Int. Conf. on Multimedia*. ACM, 2009, pp. 637–640.

[18] T.-T. Do, E. Kijak, T. Furon, and L. Amsaleg, "Deluding image recognition in SIFT-based CBIR systems," in *Proc. ACM workshop on Multimedia in Forensics, Security and Intell.* ACM, 2010, pp. 7–12.

[19] T. T. Do, E. Kijak, T. Furon, and L. Amsaleg, "Challenging the security of content-based image retrieval systems," in *Proc. IEEE Int. Workshop on Multimedia Signal Processing*, 2010, pp. 52–57.

[20] 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. on Acoustics, Speech and Signal Processing*. IEEE, 2012, pp. 1817–1820.

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

[22] 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. IEEE Int. Symp. Commun., Control, Signal Process (ISCCSP)*. IEEE, May 2012, pp. 1–5.

[23] I. Amerini, M. Barni, R. Caldelli, and A. Costanzo, "Counter-forensics of SIFT-based copy-move detection by means of keypoint classification," *EURASIP J. on Image and Video Proc.*, vol. 2013, no. 1, pp. 1–17, 2013.

[24] I. Amerini, F. Battisti, R. Caldelli, M. Carli, and A. Costanzo, "Exploiting perceptual quality issues in countering SIFT-based forensic methods," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 2664–2668.

[25] R. Caldelli, I. Amerini, and A. Costanzo, "SIFT match removal and keypoint preservation through dominant orientation shift," in *Proc. European Signal Processing Conference (EUSIPCO)*, 2015, pp. 2107–2111.

[26] J. Luo, Y. Ma, E. Takikawa, S. Lao, M. Kawade, and B. L. Lu, "Person-specific SIFT features for face recognition," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2, April 2007, pp. 593–596.

[27] Y. Han, J. Yin, and J. Li, "Human face feature extraction and recognition base on SIFT," in *Proc. IEEE Int. Symposium on Computer Science and Computational Technology (ISCSCT)*, vol. 1, Dec. 2008, pp. 719–722.

[28] T.-T. Do, E. Kijak, T. Furon, and L. Amsaleg, "Understanding the security and robustness of SIFT," in *Proc. ACM Int. Conf. on Multimedia*. ACM, 2010, pp. 1195–1198.

[29] A. Costanzo, I. Amerini, R. Caldelli, and M. Barni, "Forensic analysis of SIFT keypoint removal and injection," *IEEE Trans. on Information Forensics and Security*, vol. 9, no. 9, pp. 1450–1464, Sept. 2014.

[30] I. Amerini, M. Barni, R. Caldelli, and A. Costanzo, "SIFT keypoint removal and injection for countering matching-based image forensics," in *Proc. ACM workshop on Information hiding and multimedia security (IH&MMSec)*. ACM, 2013, pp. 123–130.

[31] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, April 2004.
[32] B. Ni, P. Moulin, and S. Yan, "Order preserving sparse coding," *IEEE Trans. on Pattern Anal. and Mach. Intell.*, vol. 37, no. 8, pp. 1615–1628, Aug 2015.
[33] J. Weickert and H. Scharr, "A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance," *J. Visual Commun. Image Represent.*, vol. 13, no. 1, pp. 103 – 118, 2002.
[34] D. Sheet, H. Garud, A. Suveer, M. Mahadevappa, and J. Chatterjee, "Brightness preserving dynamic fuzzy histogram equalization," *IEEE Trans. on Consumer Electronics*, vol. 56, no. 4, pp. 2475–2480, Nov. 2010.
[35] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics gems IV*. Academic Press Professional, Inc., 1994, pp. 474–485.
[36] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms (2008)," 2012.
[37] I. Amerini, M. Barni, R. Caldelli, and A. Costanzo, "Removal and injection of keypoints for SIFT-based copy-move counter-forensics," *EURASIP J. on Image and Video Proc.*, vol. 2013, no. 1, pp. 1–12, 2013.
[38] G. Schaefer and M. Stich, "UCID - An uncompressed colour image database," in *Proc. SPIE: Storage and Retrieval Methods and Applications for Multimedia*, vol. 5307, 2004, pp. 472–480.
[39] A. J. Fridrich, B. D. Soukal, and A. J. Lukáš, "Detection of copy-move forgery in digital images," in *Proc. Digital Forensic Research Workshop*. Citeseer, Aug. 2003.
[40] O. M. Al-Qershi and B. E. Khoo, "Passive detection of copy-move forgery in digital images: State-of-the-art," *Forensic Science International*, vol. 231, no. 1, pp. 284 – 295, 2013.
[41] D. Cozzolino, G. Poggi, and L. Verdoliva, "Efficient dense-field copy-move forgery detection," *IEEE Trans. on Inf. Forensics and Security*, vol. 10, no. 11, pp. 2284–2297, Nov 2015.
[42] D. Cozzolino, G. Poggi., and L. Verdoliva, "Copy-move forgery detection based on patchmatch," in *Proc. IEEE Int. Conf. Image Process.*, Oct 2014, pp. 5312–5316.
[43] L. Gaborini, P. Bestagini, S. Milani, M. Tagliasacchi, and S. Tubaro, "Multi-clue image tampering localization," in *Proc. IEEE Int. Workshop on Inf. Forensics and Security*, Dec 2014, pp. 125–130.
[44] S.-J. Ryu, M. Kirchner, M.-J. Lee, and H.-K. Lee, "Rotation invariant localization of duplicated image regions based on zernike moments," *IEEE Trans. on Inf. Forensics and Security*, vol. 8, no. 8, pp. 1355–1370, Aug 2013.
[45] Y. Li, "Image copy-move forgery detection based on polar cosine transform and approximate nearest neighbor searching," *Forensic Science International*, vol. 224, no. 1-3, pp. 59 – 67, 2013.
[46] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

Yuanman Li (S'15) received the B.S. in software engineering from Chongqing University, Chongqing, China, in 2012, and the M.S. degree in software engineering from University of Macau, Macau, China, in 2015. He is currently pursuing the Ph.D. degree with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau, China. His research interests include multimedia security, pattern recognition and machine learning.

Jiantao Zhou (M'11) is currently an Assistant Professor with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau. He received the B. Eng. Degree from the Department of Electronic Engineering, Dalian University of Technology, Dalian, China, in 2002, the M. Phil degree from the Department of Radio Engineering, Southeast University, Nanjing, China, in 2005, and the Ph.D. degree from the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, in 2009. He held various research positions at the University of Illinois at Urbana-Champaign, the Hong Kong University of Science and Technology, and the McMaster University. His research interests include multimedia security and forensics, and high-fidelity image compression. He was a co-author of a paper that received the Best Paper award in the IEEE Pacific-Rim Conference on Multimedia (PCM) in 2007. He holds 3 granted US patents and 2 granted Chinese patents.

An Cheng received B.S degree from Zhuhai College of Jilin University, China in 2012 and the M.S degree from University of Macau in 2015. He is currently a researcher at Meitu, Inc. His research interests include computer vison and video processing.

Xianming liu (M'12) is an Associate Professor with the Department of Computer Science, Harbin Institute of Technology (HIT), Harbin, China. He also works as a project researcher at National Institute of Informatics (NII), Tokyo, Japan, from January 2014. He received the B.S., M.S., and Ph.D degrees in computer science from HIT, in 2006, 2008 and 2012, respectively. In 2007, he joined the Joint Research and Development Lab (JDL), Chinese Academy of Sciences, Beijing, as a research assistant. From 2009 to 2012, he was with National Engineering Lab for Video Technology, Peking University, Beijing, as a research assistant. In 2011, he spent half a year at the Department of Electrical and Computer Engineering, McMaster University, Canada, as a visiting student, where he then worked as a post-doctoral fellow from December 2012 to December 2013. He has published over 40 international conference and journal publications, including top IEEE journals, such as T-IP, T-CSVT, T-IFS and T-MM; and top conferences, such as CVPR, IJCAI and DCC.

Yuan Yan Tang (S'88-M'88-SM'96-F'04) is a Chair Professor in Faculty of Science and Technology at University of Macau and Professor/Adjunct Professor/Honorary Professor at several institutes including Chongqing University in China, Concordia University in Canada, and Hong Kong Baptist University in Hong Kong. His current interests include wavelets, pattern recognition, image processing, and artificial intelligence. He has published more than 400 academic papers and is the author/coauthor of over 25 monographs/books/bookchapters. He is the Founder and Editor-in-Chief of International Journal on Wavelets, Multiresolution, and Information Processing (IJWMIP), and Associate Editors of several international journals. He is the Founder and Chair of pattern recognition committee in IEEE SMC. He has serviced as general chair, program chair, and committee member for many international conferences. Dr. Tang is the Founder and General Chair of the series International Conferences on Wavelets Analysis and Pattern Recognition (ICWAPRs). He is the Founder and Chair of the Macau Branch of International Associate of Pattern Recognition (IAPR). Dr. Y. Y. Tang is a Fellow of IEEE, and Fellow of IAPR.