JPEG Anti-Forensics With Improved Tradeoff Between Forensic Undetectability and Image Quality

Wei Fan, Kai Wang, François Cayre, and Zhang Xiong

***Abstract*— This paper proposes a JPEG anti-forensic method, which aims at removing from a given image the footprints left by JPEG compression, in both the spatial domain and DCT domain. With reasonable loss of image quality, the proposed method can defeat existing forensic detectors that attempt to identify traces of the image JPEG compression history or JPEG anti-forensic processing. In our framework, first because of a total variation-based deblocking operation, the partly recovered DCT information is thereafter used to build an adaptive local dithering signal model, which is able to bring the DCT histogram of the processed image close to that of the original one. Then, a perceptual DCT histogram smoothing is carried out by solving a simplified assignment problem, where the cost function is established as the total perceptual quality loss due to the DCT coefficient modification. The second-round deblocking and decalibration operations successfully bring the image statistics that are used by the JPEG forensic detectors to the normal status. Experimental results show that the proposed method outperforms the state-of-the-art methods in a better tradeoff between the JPEG forensic undetectability and the visual quality of processed images. Moreover, the application of the proposed anti-forensic method in disguising double JPEG compression artifacts is proven to be feasible by experiments.**

***Index Terms*— JPEG anti-forensics, DCT histogram smoothing, double JPEG compression, total variation, assignment problem.**

I. Introduction

INCREASING development of high-quality cameras and powerful photo-editing tools significantly reduces the difficulty to make visually plausible fake images. Doctored images are appearing with growing frequency, for instance, in advertising and in political and personal attacking. Doubts of the authenticity have been thrown upon digital images. Image forensics has enjoyed its popularity to restore some trust, as it serves as a passive and blind authentication technique without any *a priori* embedded information compared to digital watermarking [1]. Anti-forensics, whose objective is to mislead forensic investigators, can help researchers to study the weaknesses in existing forensic techniques for further development of trustworthy digital forensics [2].

The JPEG format is widely used as one of the most popular lossy image compression formats today, and is adopted by various digital cameras and image editing/processing software tools. On the one hand, much research has been concentrated on detecting whether an image has been JPEG compressed (or doubly JPEG compressed) for forensic purposes. On the other hand, hiding the traces left by JPEG compression is studied in the literature for anti-forensic purposes.

Fan and De Queiroz well studied the artifacts left by JPEG compression in [3], where two forensic detectors were proposed to detect JPEG artifacts in the DCT domain and in the spatial domain, respectively. The JPEG compression history of an image can also be exposed by the method in [4], which is based on JPEG error analysis. In order to conceal the JPEG compression history of digital images, Stamm *et al.* [5] pioneered the work of JPEG anti-forensics, trying to fill the gaps in the comb-like distribution of DCT coefficients in each subband. In their work, a dithering operation is proposed to conduct DCT histogram smoothing based on the Laplacian model for AC component coefficients. This dithering operation successfully fools the detector in [3] that examines DCT-domain artifacts. Targeting at the JPEG blocking artifact detector in [3] which works in the spatial domain, Stamm *et al.* later proposed to carry out a deblocking operation based on median filtering [6] after the DCT histogram smoothing [5]. Later on, researchers pointed out that the JPEG anti-forensic processing in [5] leaves footprints which can be detected by two advanced detectors [7], [8]. Another disadvantage of the method in [5] is to noticeably degrade the image visual quality, which however can be improved, to some extent, by a perceptual anti-forensic dithering method [9]. Moreover, as shown in [10], Stamm *et al.*'s

Manuscript received October 1, 2013; revised January 7, 2014 and March 19, 2014; accepted March 26, 2014. Date of publication April 17, 2014; date of current version July 1, 2014. This work was supported in part by the French ANR Estampille under Grant ANR-10-CORD-019, in part by the National Natural Science Foundation of China under Grant 61170178 and Grant 61103094, in part by the National High Technology Research and Development Program under Grant 2013AA01A601, in part by the International S&T Cooperation Program of China under Grant 2010DFB13350, and in part by the State Key Laboratory of Software Development Environment under Grant SKLSDE-2013ZX-24. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Chiou-Ting Hsu.

W. Fan is with the School of Computer Science and Engineering, Beihang University, Beijing 100191, China, and also with the GIPSA-Lab, Grenoble INP, Grenoble 38402, France (e-mail: wei.fan@gipsa-lab.grenoble-inp.fr).

K. Wang and F. Cayre are with GIPSA-Lab, CNRS UMR5216, Grenoble INP, Grenoble 38402, France (e-mail: kai.wang@gipsa-lab.grenoble-inp.fr; francois.cayre@gipsa-lab.grenoble-inp.fr).

Z. Xiong is with the School of Computer Science and Engineering, Beihang University, Beijing 100191, China (e-mail: xiongz@buaa.edu.cn).

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.2014.2317949

JPEG forgery[1] also fails to pass the detection of a machine learning based method whose feature vector is borrowed from steganalysis [11]. In order to fool some existing JPEG forensic detectors (not machine learning based) as well as to keep a high visual quality of the processed image, an approach to JPEG anti-forensics is proposed in [12], which attempts to remove the blocking artifacts in the spatial domain of the image through a variational energy minimization. It is worth noticing that some relative work can also be adopted either for JPEG anti-forensics, *e.g.*, the double JPEG compression anti-forensic method in [13], or for JPEG forensics, *e.g.*, a machine learning based steganalysis method in [14].

With the help of JPEG anti-forensics, a forger is able to falsify the image's origin or to hide evidence of tampering [6]. Different cameras often have different JPEG compression settings, *e.g.*, the customized quantization table, which can be used as a signature for image authentication [15]. After the compression history of one image is concealed through JPEG anti-forensics, the forger may compress it again using another compression setting, to falsify the original camera used to capture the image. In addition, a very likely image tampering scenario is that a forger manipulates a JPEG image (*e.g.*, cut-and-paste or splicing) using a sub-image extracted from a never compressed image or from a JPEG image of different compression setting. The resulting image may be saved using the JPEG format again. In this case, double JPEG compression artifacts may appear, at least in one part of the resulting, manipulated image [16]. For a forger, JPEG anti-forensics may be useful to hide traces of the first JPEG compression. Therefore, double JPEG compression artifacts are less likely to appear as evidence of tampering.

When JPEG compression is applied twice and no anti-forensic operation is carried out, double JPEG compression detector is a powerful tool to examine whether an image has been manipulated, or even more specifically which part of the image has been tampered. The detection of double JPEG compression artifacts can be categorized into two classes, according to whether the second JPEG compression has a grid shift with respect to the first one while applying DCT. Thus aligned double JPEG (A-DJPG) compression and non-aligned double JPEG (NA-DJPG) compression are considered separately. For detecting A-DJPG compression, the detector proposed by Pevny and Fridrich [17] is recognized as an important one in forensics, although it was initially designed for steganography. It is also the targeted detector in a recent anti-forensic work in [18]. For NA-DJPG compression detection, Bianchi and Piva [19] recently proposed a method based on an integer periodicity map built from DC coefficients. Moreover, the same authors proposed an image forgery localization method [16] via block-grained analysis of JPEG artifacts, where both A-DJPG and NA-DJPG compressions were considered.

[1]In this paper, we hereafter use "JPEG forgery" or "JPEG anti-forensic image" for referring to the post-processed JPEG image by certain anti-forensic method. The term "anti-forensic double JPEG compressed image" stands for the image which has been JPEG compressed twice, and between the two compressions, some anti-forensic operation occurs (see Sec. VI for details). Moreover, "forgery" generically means the fake image.

TABLE I
NOTATIONS

$\mathbf{I}$	original uncompressed image
$\mathbf{J}$	JPEG image compressed from $\mathbf{I}$
X	generic image pixel value matrix
Q	quantization table matrix
$(\cdot)_{i,j}$	the (i,j)-th entry of a matrix
$(\cdot)_{r,c}$	the (r,c)-th entry of an 8×8 block
$(\cdot)^l_{r,c}$	the (r,c)-th entry of the l-th 8×8 block of a matrix
$\mathbf{D}$	block DCT matrix
$\mathbf{D}^{-1}$	inverse block DCT matrix
$\mathcal{Q}(\cdot)$	block DCT coefficient quantization operator
$\mathcal{Q}^{-1}(\cdot)$	block DCT coefficient dequantization operator

In our previous JPEG anti-forensic work [12], we mainly focused on removing JPEG blocking artifacts in the spatial domain. In this paper, we extend and improve it, mainly by integrating a further step of explicit smoothing of the DCT histogram. An extended four-step procedure is proposed, which is composed of total variation (TV)-based deblocking, perceptual DCT histogram smoothing based on an adaptive local dithering signal model, second-round TV-based deblocking and *decalibration*. The effectiveness of the proposed anti-forensic method is confirmed by its undetectability against existing JPEG forensic detectors [3], [4], [7], [8], [12], in both the spatial and the DCT domains. The anti-forensic performance of our JPEG forgery is also examined using machine learning based detectors [10], [14], under a similar experimental setup to steganography. Furthermore, compared with [12], the feasibility of using the proposed method to hide evidence of JPEG re-compression is proven in this paper through the forensic testing under three double JPEG compression detectors [16], [17], [19]. Moreover, this forensic undetectability of the proposed method is achieved at the cost of very reasonable decrease of image visual quality, for disguising both the single and double JPEG compression artifacts.

The remainder of this paper is organized as follows. Sec. II briefly reviews the basics of JPEG compression and decompression. Related work on JPEG forensics, anti-forensics, and countering anti-forensics is presented in Sec. III. The proposed JPEG anti-forensic method is described in Sec. IV. Sec. V shows some experimental results with some comparisons with the prior art. In Sec. VI, the application of the proposed JPEG anti-forensic method is proven to be practical in creating anti-forensic double JPEG compressed images, with experimental comparisons with the state-of-the-art methods. Finally, concluding remarks are given in Sec. VII.

II. Basics of JPEG Compression

Without loss of generality, in this paper we consider 8-bit grayscale images. Detailed descriptions of JPEG compression can be found in [20]. Here we briefly review the process of JPEG compression and decompression. Some notations used in this paper are summarized in Table I.

The original uncompressed image $\mathbf{I}$ is firstly split into L non-overlapping 8×8 pixel value blocks. For each block, a 2-dimensional DCT is afterwards applied to obtain its

corresponding DCT coefficient block. As DCT is an orthogonal linear transform, this mapping can be modeled as a matrix multiplication $\mathbf{DI}$. The (r, c)-th $(r, c = 1, 2, \ldots, 8)$ DCT coefficient $(\mathbf{DI})^l_{r,c}$ of the l-th $(l = 1, 2, \ldots, L)$ block, is then uniformly quantized as:

$$(\mathcal{Q}(\mathbf{DI}))^l_{r,c} \doteq \text{round}\left(\frac{(\mathbf{DI})^l_{r,c}}{Q_{r,c}}\right),$$

where $Q_{r,c}$ is the (r, c)-th entry of quantization table Q, and round$(\cdot)$ is the rounding function. The resulting, quantized DCT coefficients $\mathcal{Q}(\mathbf{DI})$ are then losslessly encoded.

As to the decompression, the quantized DCT coefficient is extracted from the decoded bitstream, and then dequantized by multiplying it by the corresponding quantization step:

$$(\mathcal{Q}^{-1}(\mathcal{Q}(\mathbf{DI})))^l_{r,c} \doteq (\mathcal{Q}(\mathbf{DI}))^l_{r,c} \times Q_{r,c}.$$

The dequantized DCT coefficients are then transformed to the spatial domain by the 2-D inverse discrete cosine transform (IDCT), which can be modeled as multiplication by the 8×8 block IDCT matrix $\mathbf{D}^{-1}\mathcal{Q}^{-1}(\mathcal{Q}(\mathbf{DI}))$. At last rounding and truncation operation (denoted as $\mathcal{RT}(\cdot)$) is applied to constrain the pixel values to be integers within $[0, 255]$, and the decoded JPEG image is obtained as:

$$\mathbf{J} = \mathcal{RT}(\mathbf{D}^{-1}\mathcal{Q}^{-1}(\mathcal{Q}(\mathbf{DI}))).$$

In this paper, at the standpoint of a forger, we make the reasonable assumption that the quantized DCT coefficients $\mathcal{Q}(\mathbf{DI})$ and the quantization table Q are available to us. For example, we can use Sallee's Matlab JPEG toolbox [21] to read them from the JPEG file that we want to manipulate.

III. Prior Art

During the JPEG compression, two known artifacts appear, indicating the JPEG compression history of one image. The first one is the *quantization artifacts* in the *DCT domain*. The DCT coefficients are clustered around the integer multiples of the quantization step, leaving a comb-like distribution of DCT coefficients in each subband. The second one is the *blocking artifacts* in the *spatial domain*. There are consistent discontinuities across block borders. Both of them are traces left from an image's JPEG compression history.

A. Detecting JPEG Compression

Fan and De Queiroz [3] proposed an algorithm for maximum-likelihood estimation (MLE) of the JPEG quantization table, from a spatial-domain bitmap representation of the image. The method can also serve as a detector to classify an image as not JPEG compressed, if each entry of the estimated quantization table is either 1 or "undetermined" [3], [5].

Focusing on 8×8 block boundaries, Fan and De Queiroz [3] also proposed a JPEG blocking signature measure as:

$$K_F = \sum_k |H_I(k) - H_{II}(k)|, \tag{1}$$

where H_I and H_{II} are normalized histograms of pixel value differences across block boundaries and within the block, respectively (see [3] for details).

Luo *et al.* [4] proposed a 1-dimensional feature to distinguish JPEG images from uncompressed ones, based on the AC coefficient distribution change in the range of $(-1, 1)$ and that in the union region of $(-2, -1]$ and $[1, 2)$. They further estimated the AC component quantization step as the integer, which is greater than 2 and gives the maximum value of the DCT histogram. Experimental results show that this JPEG image detector outperforms Fan and De Queiroz's method [3].

B. JPEG Anti-Forensics

In the DCT domain, subband (r, c) contains the (r, c)-th DCT coefficient from each DCT coefficient block. For the AC component, *i.e.*, subband $(r, c) \neq (1, 1)$, the Laplacian distribution is a popular choice [22] for modeling the distribution of unquantized DCT coefficients. For the DC component, *i.e.*, subband $(1, 1)$, there is no general model representing its distribution. Nevertheless, its DCT *quantization noise* can be assumed to follow a uniform distribution [5].

In order to disguise the DCT quantization artifacts of a JPEG image, Stamm *et al.* [5] proposed to add a dithering signal to the dequantized DCT coefficients $\mathcal{Q}^{-1}(\mathcal{Q}(\mathbf{DI}))$, so that the dithered DCT coefficients approximate the estimated distribution of the unquantized ones. This JPEG anti-forensic method succeeds in fooling the quantization table estimation based detector proposed in [3].

After the dithering operation, Stamm *et al.* [6] later proposed an anti-forensic deblocking operation against the blocking artifact detector in [3], that is, the blocking signature measure K_F of Eq. (1). For a given image X, the anti-forensically deblocked image $\tilde{X}$ is obtained according to:

$$\tilde{X}_{i,j} = \text{med}_s(X_{i,j}) + w_{i,j},$$

where $\text{med}_s(\cdot)$ is the median filtering operation with local window size s, and $w_{i,j}$ is a low-power white Gaussian noise of variance σ^2.

Valenzise *et al.* proposed a perceptual anti-forensic dithering operation [9], whose resulting JPEG forgery has a higher perceptual quality than the one processed by Stamm *et al.*'s dithering method [5]. A "just-noticeable distortion" [23] criterion is adopted to control the amount of introduced distortion. A minimum-cost bipartite graph matching problem is used as the mathematical model for the adaptive insertion of the dithering signal. A greedy algorithm is implemented to get an approximate solution in order to reduce the computation cost.

Moreover, some relative techniques can also be extended for JPEG anti-forensics, *e.g.*, the Shrink-and-Zoom (SAZ) attack proposed by Sutthiwan and Shi [13], though it was initially designed for double JPEG anti-forensics. Given a JPEG image, a shrinkage (image down-scaling) operation is firstly applied; then the processed image is zoomed back to the same size as the original one, to obtain the JPEG anti-forensic image.

C. Countering JPEG Anti-Forensics

Valenzise *et al.* [7], [9], [24] claimed that the dithering signal of [5] degraded the image quality by introducing noises. Inspired by the JPEG ghosts detector [25], they designed an efficient detector against JPEG anti-forensic dithering, which

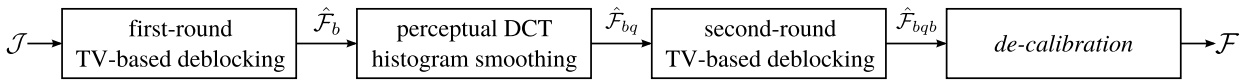


Fig. 1. The proposed JPEG forgery creation process.

examines the noisiness of re-compressed versions of the image under test. The TV of the re-compressed image (the ℓ_1 norm of the spatial first-order derivatives) [26] is employed as the image noisiness measure. For a given image, the detector re-compresses it using different quality factors $q = 1, 2, \ldots, 100$, as a function of which, $\mathrm{TV}(q)$ is computed as the total variation of the re-compressed image. The first order backward finite difference $\Delta\mathrm{TV}(q)$ is calculated as:

$$\Delta\mathrm{TV}(q) = \mathrm{TV}(q) - \mathrm{TV}(q-1),$$

with $\mathrm{TV}(0)$ prescribed to be 0. The forensic measure is:

$$K_V = \max_{q\in\{1,2,\ldots,100\}} \Delta\mathrm{TV}(q). \tag{2}$$

Lai and Böhme [8] proposed another calibration-based detector to counter Stamm *et al.*'s JPEG anti-forensic method [5], borrowing the idea of calibration from steganalysis [27]. The detector compares the variances of the corresponding subbands from a given image X and its calibrated version X_{cal}, which is obtained by cropping X by 4 pixels both horizontally and vertically. The calibrated feature K_L is then established as:

$$K_L = \frac{1}{28}\sum_{k=1}^{28}\left|\frac{\mathrm{var}(\mathbf{D}_k X) - \mathrm{var}(\mathbf{D}_k X_{cal})}{\mathrm{var}(\mathbf{D}_k X)}\right|, \tag{3}$$

where $\mathrm{var}(\cdot)$ returns the sample variance of the input vector, and $\mathbf{D}_k$ is a matrix extracting the DCT coefficients of the k-th *high-frequency subband* (as defined in [8]).

Li *et al.* [10] considered the process of creating JPEG forgery [5], [6] as data hiding, in the view of JPEG steganalysis. The anti-forensic process changes the intra- and inter-block statistics of the image, which can be measured using a group of Markov process transition probability matrices [11]. A 100-dimensional feature vector is then extracted and fed to a Support Vector Machine (SVM) for building the JPEG forensic detector. Similarly, the well-known Subtractive Pixel Adjacency Matrix (SPAM) feature vector [14] has also been used for countering JPEG anti-forensics [24].

Finally, for detecting JPEG blocking artifacts, a family of measures were built in [12]:

$$K_U^p = \left|B_{gr}^p(X) - B_{gr}^p(X_{cal})\right|, \tag{4}$$

where B_{gr}^p is the gradient aware blockiness [28], which is the normalized ℓ_p norm of the weighted gradient computed from each group of four adjacent pixel values across 8×8 block borders (see [28] for details). Note that the parameter p can vary to build a family of different measures.

IV. The Proposed Method

In this section, we propose a novel method for JPEG anti-forensics which can remove from a JPEG image both the blocking artifacts in the spatial domain and the DCT quantization artifacts in the DCT domain. Our method is able to fool existing JPEG forensic detectors, meanwhile it ensures a high visual quality of the processed image.

In practice, we find it extremely difficult to conduct a single-step attack to defeat multiple JPEG forensic detectors that work in different domains, while keeping a high image visual quality. Therefore, in this paper, we consider removing JPEG artifacts alternatively in the spatial and in the DCT domains. A similar strategy is adopted in Stamm *et al.*'s JPEG anti-forensic method [5], [6], where DCT histogram artifacts and blocking artifacts are handled separately in different domains.

As illustrated in Fig. 1, the proposed method consists of four steps for creating a JPEG forgery. The first step is TV-based deblocking in the spatial domain (to be described in Sec. IV-A). Besides the removal of JPEG blocking artifacts, another purpose of this step is to partly and plausibly fill gaps in the DCT histogram, so as to facilitate the following step of explicit histogram smoothing. Experimentally, it is necessary and beneficial to conduct this first-round deblocking, especially for a better histogram restoration in the high-frequency subbands where all DCT coefficients are quantized to zero in the JPEG image (relevant results will be presented in Sec. IV-B.4).

We found that in the deblocked image $\hat{\mathcal{F}}_b$, the comb-like DCT quantization artifacts are no longer as obvious as those in the JPEG image $\mathcal{J}$ (an example is shown in Fig. 3-(c)). Under the hypothesis that the partly recovered DCT-domain information is reliable, the next step naturally goes to further filling the remaining gaps in the DCT histogram. This leads us to the construction of an adaptive local model for the DCT coefficient distribution, with which a perceptual histogram mapping method is thereafter proposed to modify the DCT coefficients while minimizing the total SSIM (structural similarity) [29] value loss (to be described in Sec. IV-B).

The removal of the DCT quantization artifacts is at the cost of introducing a small amount of unnatural noise and blocking artifacts in the spatial domain to the output image $\hat{\mathcal{F}}_{bq}$, despite that we have tried to minimize the image quality loss. Hence, we move to the spatial domain again and conduct a second-round TV-based deblocking and regularization (to be described in Sec. IV-C). The resulting image $\hat{\mathcal{F}}_{bqb}$ is at last processed by the *decalibration* operation (to be described in Sec. IV-D) to generate the JPEG forgery $\mathcal{F}$.

A. JPEG Deblocking Using Constrained TV-Based Minimization

Researchers have investigated the problem of removing JPEG blocking artifacts. However, their efforts mainly focused on improving the image visual quality, especially for highly compressed images. Anti-forensics should take into account both the forensic undetectability and the perceptual quality. For JPEG deblocking purposes, we hereby propose a variational approach to minimize a TV-based energy consisting of a TV term and a TV-based blocking measurement term.

Inspired by [30], which aims to improve the visual quality of JPEG images compressed at low bit-rates by solving a constrained and weighted TV minimization problem, for an image X of size $H \times W$, we first define the TV term as:

$$\mathrm{TV}_b(X) = \sum_{1 \le i \le H, 1 \le j \le W} v_{i,j}, \tag{5}$$

with the variation at location (i, j) defined as:

$$v_{i,j} = \Big((X_{i-1,j} + X_{i+1,j} - 2X_{i,j})^2 + (X_{i,j-1} + X_{i,j+1} - 2X_{i,j})^2 \Big)^{1/2},$$

where $X_{i,j}$ is the value of the (i, j)-th pixel of image X. In practice, this definition of $v_{i,j}$ works well in the proposed JPEG anti-forensic framework, and we leave the comparison of different definitions of local variation as a future effort.

In order to remove the statistical traces of JPEG blocking artifacts, we define a second term which measures the JPEG blocking. The idea is very simple: it assumes that if there is no JPEG compression, statistically the energy sum of the pixel value variation along the block borders should be close to that within the block. In other words, the energy sum shall statistically remain the same no matter where the 8×8 block starts. Hence, we divide all the pixels in the image into two sets according to their positions in the block. The pixel classification strategy is illustrated in Fig. 2. Pixels at shaded locations are put into set A, while the others are put into set B. Based on this, the second energy term is defined as:

$$C(X) = \left| \sum_{X_{i,j} \in A} v_{i,j} - \sum_{X_{i,j} \in B} v_{i,j} \right|. \tag{6}$$

We also adopt a similar, yet more flexible constraint (controlled by parameter μ, a small positive number) than that in [30], with the objective to achieve a good quality of the processed image. Denote $\mathbb{M}$ as the set of integers within the range $[0, 255]$. We define the constraint image space U as:

$$U = \Big\{ X \in \mathbb{M}^{H \times W} \mid (\mathbf{D}X)^l_{r,c} \in \big[(k^l_{r,c} - \mu) Q_{r,c}, (k^l_{r,c} + \mu) Q_{r,c} \big]; r, c = 1, 2, \ldots, 8; l = 1, 2, \ldots, L \Big\}, \tag{7}$$

where $k^l_{r,c} = (\mathcal{Q}(\mathbf{DI}))^l_{r,c}$ are the quantized DCT coefficients, which can be read from JPEG image $\mathbf{J}$ using the Matlab JPEG toolbox [21]. We set this constraint space to make sure that the DCT coefficients of the processed image are within the same quantization bins (if $\mu \le 0.5$), or in the same or neighboring bins (if $\mu > 0.5$), as those of the JPEG image $\mathbf{J}$, so as to ensure an acceptable quality of the processed image.

The final constrained TV-based minimization problem is:

$$X^* = \arg\min_{X \in U} E(X) = \arg\min_{X \in U} \left(\mathrm{TV}_b(X) + \alpha C(X) \right), \tag{8}$$

where $\alpha > 0$ is a regularization parameter, balancing the two energy terms. It is easy to demonstrate that $E(X)$ is a convex function (though not differentiable) and U is a convex set

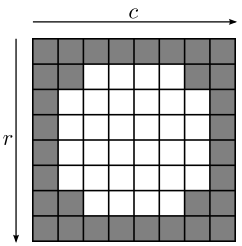


Fig. 2. Pixel classification according to its position in the 8×8 block.

[30], [31]. The optimization problem can be solved using the projected subgradient method [32] leading to the iteration:

$$X^{(k+1)} = P_U \left(X^{(k)} - t \times g(X^{(k)}) \right), \tag{9}$$

where $X^{(k)}$ is the processed image at the k-th iteration (note that, $X^{(0)}$ is the given JPEG image), $t > 0$ is the step size, $g(X)$ is a subgradient [32] of $E(X)$, and P_U is the projection operator onto space U that will be explained later.

The regularization parameter α in Eq. (8), the step size t in Eq. (9), and μ in Eq. (7) are parameters that we can adjust. In practice, we set $\alpha = 1.5$, as it achieves a good tradeoff between the visual quality and the DCT histogram restoration quality of the processed image (see the Supplementary Material, which can be downloaded online[2], for details and experimental results). The step size is set to $t = 1/k$ at the k-th iteration, following [30]. As to the setting of the convex set U, here we set $\mu = 0.5$, which strictly constrains the processed DCT coefficient to stay within the same quantization bin as its original value. Besides the visual quality control, this is also favorable for the perceptual DCT histogram smoothing to be described in Sec. IV-B, where the processed DCT coefficient is also constrained to stay within its original quantization bin. The projection operator P_U works as follows: once a DCT coefficient under processing goes outside the quantization bin of its original value, it will be modified back to a random value uniformly distributed within the quantization bin. In the spatial domain the resulting pixel values will at last be rounded and truncated to integers in the range $[0, 255]$.

Instead of waiting for the convergence of the optimization problem, we have a different strategy to select the candidate deblocked image, which may not be the solution X^* in Eq. (8). The selection is guided by the blocking signature measure K_F in Eq. (1). Indeed, in practice we found that for uncompressed images, the output of K_F has a smaller standard deviation than another blocking signature K^p_U in Eq. (4). Its detection ability is also proven to be stronger than K^p_U when tested on JPEG images (see Table V in Sec. V-B). Therefore, although it is difficult to include K_F in our optimization framework as it is histogram based, it would be reasonable to use K_F to guide the selection of the deblocked result. Experimentally, we run 50 iterations, and choose the resulting image giving the smallest K_F value as the final result. This provides us with a satisfying intermediate image $\hat{\mathcal{F}}_b$.

[2]http://www.gipsa-lab.fr/~wei.fan/documents/AFJPG-TIFS14.tar.gz
The code of the proposed method is also included in the package.

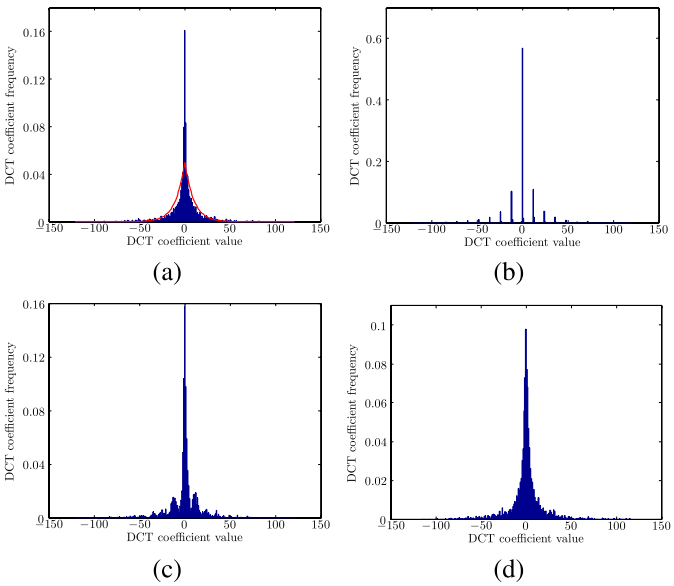


Fig. 3. (a) is the DCT histogram of subband $(2, 2)$ from an example genuine, uncompressed UCID-v2 [33] image, and the red curve is the fitting result using the discrete Laplacian distribution model. Then the image is JPEG compressed with quality factor 50, and the proposed JPEG anti-forensic method is applied. (b), (c), and (d) are the corresponding DCT histograms of (a) in the JPEG image, after the first-round TV-based deblocking, and after the adaptive local dithering signal is injected (as described in Sec. IV-B), respectively.

B. Perceptual DCT Histogram Smoothing

After JPEG image $\mathcal{J}$ has been processed using the TV-based deblocking method, the gaps in the DCT domain have been partly filled in the obtained image $\hat{\mathcal{F}}_b$ (an example DCT histogram is shown in Fig. 3-(c)). We have confirmed the effectiveness of the TV-based deblocking in fooling existing JPEG forensic detectors in our previous work [12], yet with a different parameter setting of μ in Eq. (7). However, the periodicity of the DCT histogram in $\hat{\mathcal{F}}_b$ may still exist, which might be utilized by JPEG forensic detectors. This issue will be further discussed in the application of JPEG anti-forensics in Sec. VI-A. In order to achieve a better forensic undetectability, it is necessary to fill the gaps left in the DCT histogram of $\hat{\mathcal{F}}_b$. In this section, we propose a perceptual DCT histogram smoothing method. The partly recovered information in the DCT domain of $\hat{\mathcal{F}}_b$ will help us to build an adaptive local dithering signal model based on both the Laplacian distribution and the uniform distribution for a better goodness-of-fit.

1) Disadvantages When Using the Global Laplacian Model: Constructing the DCT histogram is a common practice for studying the image statistics in the DCT domain. In this paper, all the DCT histograms are constructed using *integers* as the bin centers. For subband (r, c) of image X, the normalized DCT histogram is therefore constructed as:

$$H^X_{r,c}(k) = \frac{1}{L}\sum_{l=1}^{L}\delta\Big(\text{round}\big((\mathbf{D}X)^l_{r,c}\big) - k\Big), \quad k \in \mathbb{Z} \qquad (10)$$

where δ is the indicator function: $\delta(x) = 1$ if and only if $x = 0$, otherwise $\delta(x) = 0$.

For modeling the DCT coefficients in AC components, the Laplacian distribution is the dominant choice balancing the model simplicity and the fidelity to real data [22], which is also the basic assumption of Stamm *et al.*'s dithering-based JPEG anti-forensics [5]. Fig. 3-(a) shows an example DCT histogram from a genuine, uncompressed UCID-v2 [33] image, and the fitting result using the discrete Laplacian distribution:

$$P(Y = y) = \begin{cases} 1 - e^{-\lambda/2} & \text{if} \quad y = 0 \\ e^{-\lambda|y|}\sinh(\lambda/2) & \text{if} \quad y \in \mathbb{Z}, y \neq 0 \\ 0 & \text{otherwise,} \end{cases} \qquad (11)$$

where the parameter λ can be estimated using MLE [34]. From Fig. 3-(a), we can see that the Laplacian distribution may not always be a good model for a precise description of real data.

Furthermore, as known, the kurtosis (different from *excess kurtosis* including a -3 term) of the Laplacian distribution is a constant 6. We calculated the kurtosis of all the AC components for each UCID-v2 image [33], and 93.68% of them have values higher than 6. The average kurtosis is 19.99, much higher than 6, which indicates that the actual distribution of DCT coefficients usually has a much higher peak. This also partly explains the fitting problem of the DCT histogram using the Laplacian model in Fig. 3-(a).

Moreover, Robertson and Stevenson [35] pointed out that for quantized DCT coefficients that are observed to be zero, *i.e.*, the DCT coefficients in the quantization bin 0 satisfying $(\mathcal{Q}(\mathbf{D}X))^l_{r,c} = 0$, the Laplacian model indeed works well. However, for the other DCT coefficients in the quantization bin $b \neq 0$ satisfying $(\mathcal{Q}(\mathbf{D}X))^l_{r,c} = b$, it appears that the uniform model fits better to real data than the Laplacian model.

2) Adaptive Local Dithering Signal Model: Based on the above analysis, we hope to build a model having a better goodness-of-fit than the global Laplacian model. Comparing an example histogram shown in Fig. 3-(c) with that in -(b), we notice that the DCT-domain information has been partly recovered in $\hat{\mathcal{F}}_b$, with the help of which we will be able to build an adaptive local dithering signal model in this section.

We still use X to denote a generic image pixel value matrix, however X now contains pixel values of the post-processed JPEG image $\hat{\mathcal{F}}_b$ using the TV-based deblocking method in Sec. IV-A. For a given DCT subband (r, c) of $\hat{\mathcal{F}}_b$, with $Q_{r,c}$, the DCT coefficients $(\mathbf{D}X)^l_{r,c}$ (generically denoted as Y') are quantized and then dequantized to obtain coefficients $(\mathcal{Q}^{-1}(\mathcal{Q}(\mathbf{D}X)))^l_{r,c}$ (generically denoted as Y). The goal is to add a dithering signal N in such a way that the dithered DCT coefficient:

$$Z = Y + N$$

has no comb-like DCT quantization artifacts while its distribution is to some extent close to that of Y'. In this section, we are to find a proper distribution for the dithering signal N.

We firstly consider the AC component. Without special explanation, the DCT coefficients mentioned in this section are all from the AC component subbands. Inspired by Robertson and Stevenson's work about the suitability of the Laplacian model and the uniform model for different parts of the DCT histogram [35], we propose an *adaptive local* dithering signal model based on the combination of the Laplacian

distribution and the uniform distribution, with the appropriate parameter tuned in each quantization bin.

We denote $B_{r,c}^{-} = \min\left((\mathcal{Q}(\mathbf{D}X))_{r,c}^{l}\right)$ as the non-empty quantization bin with the smallest bin center value, whereas $B_{r,c}^{+} = \max\left((\mathcal{Q}(\mathbf{D}X))_{r,c}^{l}\right)$ as the non-empty quantization bin with the largest bin center value. We build the dithering signal model through one quantization bin by another, starting from quantization bin $b = 0$.

Given quantization bin b, we try to seek for parameter λ_b of the Laplacian distribution by solving the following constrained weighted least-squares fitting problem:

$$\lambda_b = \underset{\lambda_b^{-} \le \lambda \le \lambda_b^{+}}{\arg\min} \sum_{k=B_{r,c}^{-}Q_{r,c}-\lfloor \frac{Q_{r,c}}{2} \rfloor}^{B_{r,c}^{+}Q_{r,c}+\lfloor \frac{Q_{r,c}}{2} \rfloor} w_k \times \left(H_{r,c}^{X}(k) - P(Y=k)\right)^2, \tag{12}$$

where $H_{r,c}^{X}$ and P are defined in Eqs. (10) and (11), respectively. The fitting problem in Eq. (12) means, that we wish to find a local Laplacian distribution which is still close to the corresponding distribution in $\hat{\mathcal{F}}_b$. We set $w_k = \left(|\text{round}(\frac{k}{Q_{r,c}}) - b| + 1\right)^{-1}$ as the weight for the deduction of λ_b, which emphasizes the importance of the DCT coefficients from the current quantization bin b for the fitting. We also studied some other settings of w_k, *e.g.*, the same function as the one used above but with different powers, the Gaussian function, *etc*. We found that in practice different settings of w_k have minor impact on the histogram restoration quality, and that the current setting yields slightly better results. Moreover, λ_b^{-} and λ_b^{+} in Eq. (12) are the lower and upper bounds of the parameter λ. If λ_b^{-} and λ_b^{+} are well defined, then the fitting problem can be established and λ_b *can be found* by solving Eq. (12); otherwise, the fitting problem cannot be established and we say that λ_b *cannot be found*. Before we describe how the searching of the two bounds λ_b^{-} and λ_b^{+} is performed, we first explain the models in use for each quantization bin b.

If the parameter λ_b can be found for quantization bin b by solving a well-defined fitting problem Eq. (12), the Laplacian model will be used. In this case, we follow Stamm *et al.*'s dithering signal model [5]. The distribution of the dithering signal N is given by (replacing λ by the actual value of λ_b):

$$P(N=n|Y=0) = \begin{cases} c_0 e^{-\lambda|n|} & \text{if } -\frac{Q_{r,c}}{2} < n < \frac{Q_{r,c}}{2} \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

$$P(N=n|Y=y, y>0) = \begin{cases} c_1 e^{-\lambda n} & \text{if } -\frac{Q_{r,c}}{2} \le n < \frac{Q_{r,c}}{2} \\ 0 & \text{otherwise,} \end{cases} \tag{14}$$

$$P(N=n|Y=y, y<0) = \begin{cases} c_1 e^{\lambda n} & \text{if } -\frac{Q_{r,c}}{2} < n \le \frac{Q_{r,c}}{2} \\ 0 & \text{otherwise,} \end{cases} \tag{15}$$

with $c_0 = \frac{\lambda}{2}(1 - e^{-\lambda Q_{r,c}/2})^{-1}$ and $c_1 = \lambda e^{-\lambda Q_{r,c}/2}(1 - e^{-\lambda Q_{r,c}})^{-1}$. Let P_m^o and P_m^e, two functions of λ, denote the probability mass function (p.m.f.) of the *rounded* dithering signal when $Q_{r,c}$ is an odd number and an even number, respectively. The domain of the p.m.f. is the integer set $\{-\lfloor \frac{Q_{r,c}}{2} \rfloor, -\lfloor \frac{Q_{r,c}}{2} \rfloor + 1, \ldots, \lfloor \frac{Q_{r,c}}{2} \rfloor\}$. Due to space limit, we omit the equations of the p.m.f. here. Interested readers could refer to the Supplementary Material. The p.m.f. will be used later for the searching of λ_b^{-} and λ_b^{+} of Eq. (12).

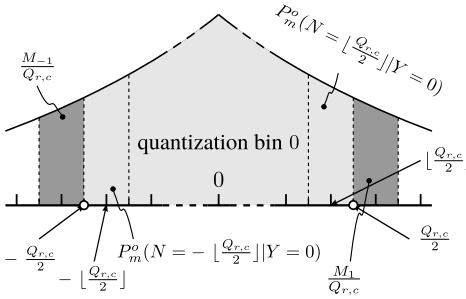


Fig. 4. Example to illustrate the constraint used for the searching of λ_b^{+} when $Q_{r,c}$ is an odd number: The probability for the leftmost (or the rightmost) integer bin of the quantization bin $b = 0$ should be no smaller than either that of the rightmost integer bin in the quantization bin $b = -1$ or that of the leftmost integer bin in the quantization bin $b = 1$.

As to the quantization bin b, where λ_b cannot be found, the uniform model will be used instead. In this case, the dithering signal N is generated according to:

$$P(N=n|Y=y) = \begin{cases} \frac{1}{Q_{r,c}} & \text{if } n \in \mathcal{N} \\ 0 & \text{otherwise,} \end{cases} \tag{16}$$

where $\mathcal{N} = (-\frac{Q_{r,c}}{2}, \frac{Q_{r,c}}{2})$ when $y = 0$, $\mathcal{N} = [-\frac{Q_{r,c}}{2}, \frac{Q_{r,c}}{2})$ when $y > 0$, and $\mathcal{N} = (-\frac{Q_{r,c}}{2}, \frac{Q_{r,c}}{2}]$ when $y < 0$.

Now, we go back to the searching of the bounds λ_b^{-} and λ_b^{+} used in Eq. (12). We start from the center of the DCT histogram, *i.e.*, quantization bin $b = 0$. We set $\lambda_b^{-} = 10^{-3}$ according to Valenzise *et al.*'s statement [9] that the parameter λ of the Laplacian model usually takes values between 10^{-3} and 1 for natural images. The constraint used in the searching is based on the observation that in the distribution of DCT coefficients, the probability decreases when the coefficient magnitude increases. As an example and as illustrated in Fig. 4, when $Q_{r,c}$ is an odd number, we constrain that the probability of DCT coefficient falling in the leftmost integer bin $k = -\lfloor \frac{Q_{r,c}}{2} \rfloor$ (or the rightmost integer bin $k = \lfloor \frac{Q_{r,c}}{2} \rfloor$) of the quantization bin $b = 0$ should be no smaller than either that in the rightmost integer bin of the quantization bin $b = -1$ or that in the leftmost integer bin of the quantization bin $b = 1$. For the moment, in the neighboring quantization bins -1 and 1, the DCT coefficients are assumed to follow a uniform distribution. Then λ_b^{+} is determined by solving Eq. (17),

$$\lambda_b^{+} = \underset{10^{-3} \le \lambda \le 1}{\arg\max}\ \lambda, \text{subject to}: P_m^o\left(N = \left\lfloor \frac{Q_{r,c}}{2} \right\rfloor | Y = 0\right) \times M_0 \ge \max\left(\frac{M_{-1}}{Q_{r,c}}, \frac{M_1}{Q_{r,c}}\right) \tag{17}$$

where M_0, M_{-1}, M_1 are respectively the approximate probabilities of DCT coefficient falling in quantization bins $0, -1, 1$, which are estimated directly from the constructed histogram. The searching for λ_b^{+} can be done using a numerical method. A set of numbers are uniformly sampled from the interval $[10^{-3}, 1]$. Given each number in this set as the parameter λ, P_m^o is calculated. Therefore, λ_b^{+} is chosen as the largest number satisfying constraints in Eq. (17). When

Algorithm 1 Adaptive Local Dithering for AC Components

Require: Y
Initialization: $Z = Y$
for $b = 0, 1, 2, \ldots, B^+_{r,c}, -1, -2, \ldots, B^-_{r,c}$ **do**
  Search for λ^+_b and λ^-_b (*e.g.*, using Eq. (17))
  if λ^+_b and λ^-_b are well defined **then**
    $\lambda_b \leftarrow$ Eq. (12)
    Generate N using Eq. (13)/(14)/(15)
    $Z \leftarrow Z + N$
  else
    Generate N using Eq. (16)
    $Z \leftarrow Z + N$
  end if
end for
return Z

$Q_{r,c}$ is an even number, the procedure to find λ^+_b is similar yet slightly different. Detailed information can be found in the Supplementary Material.

For quantization bins $b = 1, 2, \ldots, B^+_{r,c}$, we adopt a similar procedure to obtain λ^-_b and λ^+_b. The constraint is that the probability of coefficients falling in the leftmost integer bin of quantization bin $b \geq 1$ should be no bigger than that in the rightmost integer bin of quantization bin $b - 1$, meanwhile the probability of coefficients falling in the rightmost integer bin of quantization bin b should be no smaller than that in the leftmost integer bin of quantization bin $b + 1$. The distribution in the quantization bin $b - 1$ is already defined at that moment, and for the quantization bin $b + 1$ we assume the uniform distribution. Then λ^-_b and λ^+_b are chosen as the smallest and largest number in $[10^{-3}, \lambda_{b-1}]$ satisfying the constraint, respectively. However, if λ^-_b and λ^+_b cannot be found, the uniform model in Eq. (16) will be used as the dithering signal model for the current and following quantization bin(s). For quantization bins $b = -1, -2, \ldots, B^-_{r,c}$, a similar searching procedure for the values of λ^-_b and λ^+_b is applied (see the Supplementary Material for details).

When we have well defined λ^-_b and λ^+_b values, the parameter λ_b of the Laplacian model is obtained by solving the minimization problem in Eq. (12), and the dithering signal N can be thereafter generated according to Eqs. (13)-(15); otherwise, the uniform model in Eq. (16) is used. Algorithm 1 summarizes the proposed adaptive local dithering procedure for the AC components.

Although we adopt Stamm *et al.*'s dithering signal model, note that their model is rather *global* and our model is *local*. In Stamm *et al.*'s model, once the parameter λ of the Laplacian distribution is estimated for a subband, it will be used for all the quantization bins of this subband. However, in our model, it is considered *locally* for each quantization bin b. We tune parameter λ_b of the Laplacian model for different quantization bins if it can be found; otherwise, the uniform model will be used instead. Moreover, in Stamm *et al.*'s method, MLE is used to estimate λ from the JPEG image. However, in the proposed method, λ_b for each quantization bin is obtained via weighted least-squares fitting using the post-processed image $\hat{\mathcal{F}}_b$ where the DCT-domain information is partly recovered. We will show in Sec. V-A that our method leads to a better restoration of the DCT histogram of the original, uncompressed image.

Because there is no general model for the DC component, we use the uniform model for all the quantization bins to generate the dithering signal N according to Eq. (16).

For each quantization bin, the dithering signal generated by numerical sampling of a given probability distribution function can reproduce the natural, fine-grained details in the DCT histogram. An example result is shown in Fig. 3-(d).

3) DCT Histogram Mapping: Now we can generate the dithered signal Z using the adaptive local dithering signal model. However we cannot use Z directly as the altered DCT coefficients by adding the dithering signal N *randomly* to Y, without any consideration of the image spatial-domain information: the processed image will suffer from low visual quality as Stamm *et al.*'s JPEG forgery does [5]. A different strategy is adopted here. We will try to move the distribution of Y' (*i.e.*, the distribution of DCT coefficients of $\hat{\mathcal{F}}_b$) towards that of Z, while minimizing the introduced distortion in the spatial domain, by solving an assignment problem whose cost function is defined as the total perceptual quality loss.

We still consider the DCT coefficients in each quantization bin *individually*. In this section, all the DCT coefficients mentioned are in a single quantization bin b of subband (r, c). A classical assignment problem can be established as follows. Let O^b denote the set of DCT coefficients in quantization bin b from Y' which are to be modified, and T^b is used to denote the set of target DCT coefficient values from Z $(= Y + N)$ falling in the quantization bin b. O^b and T^b are of equal size. The weight function $W : O^b \times T^b \rightarrow \mathbb{R}$ is defined as the SSIM value loss due to the coefficient modification, compared with the currently achieved processed image from the last solved assignment problem (or $\hat{\mathcal{F}}_b$ at the very beginning). Our goal is to find a bijection $f : O^b \rightarrow T^b$ such that the cost function:

$$\sum_{o \in O^b} W(o, f(o)) \tag{18}$$

is minimized. This problem can be solved using the well-known Hungarian algorithm[3] [36]. The solution of the assignment problem can be found in $O(D^3)$ time, where D is the dimension of the problem. If D is small, the problem however can be solved within a reasonable time. The setting of D will be further discussed in Sec. V-A.

In order to save the computation cost, we hereby propose three strategies to simplify the building and the solving of this assignment problem. Firstly, not all but only part of the DCT coefficients in Y' are to be modified, so that the dimension of the assignment problem can be largely reduced. For choosing the DCT coefficients to put into O^b, first we compare the *unnormalized* DCT histogram (using *integers* as bin centers) of Y', denoted as h^b_o, and that of Z, denoted as h^b_t. Integers $bQ_{r,c} - \lfloor \frac{Q_{r,c}}{2} \rfloor, bQ_{r,c} - \lfloor \frac{Q_{r,c}}{2} \rfloor + 1, \ldots, bQ_{r,c} + \lfloor \frac{Q_{r,c}}{2} \rfloor$ are all the possible rounded DCT coefficient values in h^b_o and h^b_t. It is obvious that the dithering process in Sec. IV-B.2 ensures that the two histograms in comparison have the same number of

[3]We use the Matlab code from: http://www.mathworks.fr/matlabcentral/fileexchange/20652-hungarian-algorithm-for-linear-assignment-problems-v2-3

DCT coefficients. We compute their difference histogram as:

$$h_d^b(k) = h_o^b(k) - h_t^b(k),$$
$$k = bQ_{r,c} - \lfloor \frac{Q_{r,c}}{2} \rfloor, \ldots, bQ_{r,c} + \lfloor \frac{Q_{r,c}}{2} \rfloor.$$

In order to reduce the dimensionality of the assignment problem, the first strategy is that we do not modify the coefficients in h_o^b that are already in the *correct* integer bin with respect to h_t^b. More precisely, for each integer bin k with $h_d^b(k) \leq 0$, the corresponding DCT coefficients are left unchanged; and for each integer bin k with $h_d^b(k) > 0$, instead of putting all the $h_o^b(k)$ coefficients in O^b, we only put $h_d^b(k)$ ($< h_o^b(k)$) coefficients into O^b while leaving the other $h_t^b(k)$ coefficients unchanged. In general, this will yield suboptimal results, however the advantage is that it will largely reduce the computation cost with still satisfactory final results. In order to control the distortion in the spatial domain, it is better to choose the DCT coefficient whose corresponding spatial-domain 8×8 block is less sensitive to noise for being altered. Using a similar strategy with [37], here we adopt the SSIM index [29] to compute a similarity map between the currently achieved image and $\hat{\mathcal{F}}_b$. The DCT coefficients whose spatial-domain 8×8 block has a higher SSIM index value are chosen to be modified and put into set O^b. Note that at the very beginning of the procedure, the SSIM index cannot be computed because the currently achieved image is exactly $\hat{\mathcal{F}}_b$. In this case, we calculate the local variance instead. We also have to reduce the dimensionality of the target value set T^b so that O^b and T^b are of equal size. To this end, for each k with $h_d^b(k) < 0$, in Z we *randomly* choose $-h_d^b(k)$ (> 0) values among the DCT coefficients whose rounded values are k, and put these values into T^b.

The second strategy to simplify the building of the assignment problem in Eq. (18) is to speed up the calculation of the weight function W. Normally, for each DCT coefficient o in O^b, all the possible modifications to values in T^b should be enforced to calculate the cost. Yet, this can be simplified by only computing the SSIM value loss when the DCT coefficient o in O^b is changed to a value in T^b which is the farthest from o. The linear interpolation is afterwards used to estimate the cost for all the other possible modifications in T^b.

The last strategy is to randomly split the simplified assignment problem into several smaller ones of lower dimensionality which can be solved in a reasonable time. This will be further discussed with experimental results in Sec. V-A, where we also show that the adoption of all these strategies will not impair the visual quality of the obtained image, despite the fact that these strategies lead to suboptimal solutions.

After solving a simplified assignment problem for each quantization bin, we are able to smooth the DCT histogram with minimum introduced distortion in the spatial domain. The created intermediate image is denoted as $\hat{\mathcal{F}}_{bq}$.

4) The Necessity of the First-Round TV-Based Deblocking: As one may have noticed, it is possible to perform the perceptual DCT histogram smoothing described in this section directly on the JPEG image, without the application of the first-round TV-based deblocking described in Sec. IV-A. Here, $\hat{\mathcal{F}}_q$ denotes the image obtained using the proposed perceptual DCT histogram smoothing directly from the JPEG image. In this paper, our large-scale test is carried out on 1338 images of size 512×384 from the UCID-v2 corpus [33]. Without loss of generality, only the luminance component (extracted using the MATLAB function *rgb2ycbcr*) is considered. Here, for comparing $\hat{\mathcal{F}}_{bq}$ and $\hat{\mathcal{F}}_q$, we randomly select 92 UCID-v2 images, and we call the smaller image dataset UCID92. Each UCID92 image is compressed with quality factor selected from $\{50, 51, \ldots, 95\}$, and every two images have the same factor.

TABLE II
IMAGE QUALITY AND KL DIVERGENCE (WITH THE UNCOMPRESSED IMAGE AS THE REFERENCE) COMPARISON ACHIEVED ON UCID92 AFTER THE PERCEPTUAL DCT HISTOGRAM SMOOTHING, WITH AND WITHOUT THE FIRST-ROUND TV-BASED DEBLOCKING

	PSNR	SSIM	KL divergence-1	KL divergence-2
$\hat{\mathcal{F}}_{bq}$	36.4194	0.9864	0.0891	0.0979
$\hat{\mathcal{F}}_q$	36.1610	0.9877	0.1262	0.2468

PSNR (Peak Signal to Noise Ratio), SSIM are adopted as the image quality evaluation metrics. In order to give a quantitative evaluation of the DCT histogram, we use the Kullback-Leibler (KL) divergence as the difference measure between the histogram of the uncompressed image and the one in the processed image. A smaller value of KL divergence means a better resemblance between the two compared histograms. Table II reports the average PSNR, SSIM, and KL divergence values of $\hat{\mathcal{F}}_{bq}$ and $\hat{\mathcal{F}}_q$. All the metric values are computed using the original uncompressed image as the reference. The difference between the two kinds of KL divergences is that the first one is averaged over the subbands where not all the DCT coefficients are quantized to 0 in the original JPEG image, whereas the other is averaged over the rest of the subbands. The lower KL divergence value of $\hat{\mathcal{F}}_{bq}$ than that of $\hat{\mathcal{F}}_q$ demonstrates that with the partly recovered DCT-domain information in $\hat{\mathcal{F}}_b$ we are able to achieve a more accurate DCT histogram restoration, especially for the subbands where all the DCT coefficients are quantized to 0 in JPEG images.

Moreover, $\hat{\mathcal{F}}_b$ also helps us to reduce the dimensionality of the simplified assignment problem during the DCT histogram mapping described in Sec. IV-B.3. The reason is that in the first strategy of simplifying the assignment problem, more DCT coefficients in $\hat{\mathcal{F}}_b$ will already be in the correct integer bin than those in the JPEG image.

Furthermore, $\hat{\mathcal{F}}_{bq}$ achieves a slightly higher PSNR value, but slightly lower SSIM value, than $\hat{\mathcal{F}}_q$. Considering the results of both metrics, the two kinds of images have comparable visual qualities. Another advantage of the TV-based deblocking is the removal of JPEG blocking artifacts. It is therefore necessary to conduct the first-round TV-based deblocking for a better tradeoff between the visual quality and the histogram restoration quality of the processed image.

C. Second-Round TV-Based Deblocking

In the perceptual DCT histogram smoothing, although we have tried to modify the DCT coefficients while minimizing the spatial-domain distortion, there must be some unnatural noise and blocking artifacts introduced in $\hat{\mathcal{F}}_{bq}$. Hence, we

focus on the spatial domain again and propose to apply the second-round TV-based deblocking and regularization.

The procedure is basically the same as that in Sec. IV-A, yet with some modifications to the parameter setting. Since the JPEG blocking artifacts presented in $\hat{\mathcal{F}}_{bq}$ are not as serious as those in $\mathcal{J}$, hence we lower the parameters α and t for a milder JPEG deblocking. We set $\alpha = 0.9$, and the step size $t = 1/(k+1)$ at the k-th iteration. As to the setting of the convex set U, here we set $\mu = 1.5$, which constrains that the processed DCT coefficient should stay within the same or the neighboring quantization bins as its original value. Once a processed coefficient goes outside of the constrained range, the projection operator P_U modifies its value back to a random value uniformly distributed in the original quantization bin. This can avoid strong DCT histogram shape modification by the TV-based deblocking and prevent the emergence of new DCT quantization artifacts. Using these empirical parameter settings, experimentally we can achieve satisfactory results considering both the forensic undetectability and the visual quality of the processed image.

We also observed that in practice, the TV-based deblocking might interfere with the output of the quantization table estimation based detector [3], especially in the high-frequency subbands. The image tends to be over-smoothed by the TV-based regularization. As analyzed in [38], the quantization table estimation based detector will detect one DCT subband as quantized by 3 instead of 1, when the probability of DCT coefficients whose rounded values are integer multiples of 3 (denoted as P_3) is higher than 67.28%. This frequently happens in the high-frequency subbands of relatively smooth images, which has a high frequency of DCT coefficients with rounded value 0.

In order to tackle this problem, we propose to introduce a slight perturbation to the DCT coefficients in the high-frequency subbands which have a high value of P_3 during each optimization iteration. Considering the influence of the subgradient method, the DCT coefficients projection of P_U, and the pixel value rounding and truncation in the spatial domain, for a given DCT subband, if $P_3 > 60\%$, we modify part of the DCT coefficients whose rounded values are 0 to the integer bins $-5, -4, \ldots, -1, 1, 2, \ldots, 5$. The DCT coefficient whose corresponding spatial-domain block has a higher tolerance of distortion (measured by the SSIM index comparing the currently achieved image and the deblocked image in the last subgradient iteration, or $\hat{\mathcal{F}}_{bq}$ at the very beginning) will be modified first to a DCT coefficient which has a bigger rounded amplitude. The modification constrains that the relative probability of integer bins $-5, -4, \ldots, -1, 1, 2, \ldots, 5$ should stay unchanged. The modification stops once P_3 reaches 50%.

In order to avoid over-smoothing of the image, a random threshold for each image is drawn from the distribution of the K_F values for genuine, uncompressed images, and the iteration stops once the K_F value in Eq. (1) drops below it. Otherwise, if this cannot be achieved within 30 iterations, the resulting intermediate image with the smallest K_F output is chosen as the final result. The created image after this step is denoted as $\hat{\mathcal{F}}_{bqb}$.

D. Decalibration

For $\hat{\mathcal{F}}_{bqb}$, all the existing detectors seems to be well fooled except the calibrated feature based detector [8]. In fact the calibrated feature value, *i.e.*, K_L of Eq. (3), has also been significantly decreased. However, for genuine, uncompressed images, this feature value is highly condensed in an interval of very small values. It is hard to further decrease this value by performing deblocking, while keeping good visual quality.

In this section, we will directly optimize an energy function which is very close to Eq. (3) for *decalibration* purposes. The minimization problem is formulated as:

$$X^* = \arg\min_X \sum_{k=1}^{28} \left| \mathrm{var}(\mathbf{D}_k X) - \mathrm{var}(\mathbf{D}_k X_{cal}) \right|, \qquad (19)$$

which can also be solved using the subgradient method [32].

As we almost directly minimize the calibrated feature value, we are able to obtain very small K_L values when converging to X^* in Eq. (19). In order to fool the detector, a random threshold for each image is drawn from the distribution of the calibrated feature values for genuine, uncompressed images, and the iteration stops once the K_L value drops below it. After decalibration, the JPEG forgery $\mathcal{F}$ is obtained.

V. Experimental Results of JPEG Anti-Forensics

For a given JPEG image $\mathcal{J}$ compressed from $\mathcal{I}$, six kinds of JPEG forgeries and two kinds of intermediate results are created as follows:

- $\mathcal{F}_{S_q}$, with the application of Stamm *et al.*'s DCT histogram smoothing method [5];
- $\mathcal{F}_{S_q S_b}$, with Stamm *et al.*'s dithering signals [5] added on $\mathcal{J}$ first, and then Stamm *et al.*'s deblocking operation is applied, with parameters $s = 3$ and $\sigma^2 = 2$ as in [6];
- $\mathcal{F}_V$, with the application of Valenzise *et al.*'s perceptual anti-forensic dithering method [9];
- $\mathcal{F}_{Su}$, with the application of the SAZ attack [13];
- $\mathcal{F}_F$, with the application of the TV-based anti-forensic method [12];
- $\mathcal{F}$, with the application of the proposed four-step JPEG anti-forensic method;
- $\hat{\mathcal{F}}_{bq}$, with the application of the proposed first-round TV-based deblocking and the perceptual DCT histogram smoothing method, the maximum dimension is set to be 200 when solving the assignment problem;
- $\hat{\mathcal{F}}'_{bq}$, the creation of which is the same as $\hat{\mathcal{F}}_{bq}$, except that there is no limit set for the maximum dimension when solving the assignment problem.

We compress each genuine, uncompressed UCID-v2 image with a quality factor randomly selected in $\{50, 51, \ldots, 95\}$, from which different kinds of JPEG forgeries are created. Among all the 1338 UCID-v2 images, we randomly select 669 images for testing and the other half of the images are left for training the SVM-based detectors [10], [14], [17] (to be described in Sec. V-B and the Supplementary Material). We call the testing dataset UCIDTest and the training dataset UCIDTrain.

(a) (b) (c) (d)

Fig. 5. Example results (close-up images) around the shoulder of Lena of $\hat{\mathcal{F}}_{bq}$ compared with $\mathcal{J}$, $\mathcal{F}_S$, and $\mathcal{F}_V$, where $\mathcal{J}$ is compressed with quality factor 50. Their SSIM values (with $\mathcal{I}$ as the reference) are: (a) 0.9809, (b) 0.9509, (c) 0.9610, and (d) 0.9731. We can see that less noise is introduced in $\hat{\mathcal{F}}_{bq}$ especially in the relatively smooth area of the image.

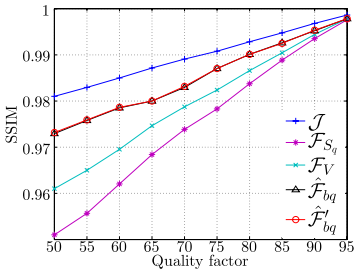


Fig. 6. SSIM values achieved by $\mathcal{J}$, $\mathcal{F}_{S_q}$, $\mathcal{F}_V$, $\hat{\mathcal{F}}_{bq}$, and $\hat{\mathcal{F}}'_{bq}$, with $\mathcal{I}$ as the reference, when tested on the classical test image "Lena". The $\hat{\mathcal{F}}_{bq}$ plot is almost under that of $\hat{\mathcal{F}}'_{bq}$.

TABLE III
THE DIFFERENCE OF THE KL DIVERGENCE BETWEEN $\mathcal{I}$ AND $\mathcal{F}_{S_q}$, AND THAT BETWEEN $\mathcal{I}$ AND $\hat{\mathcal{F}}_{bq}$ FOR ALL 64 DCT SUBBANDS. THE AVERAGE DIFFERENCE VALUE OVER ALL SUBBANDS IS 0.0903. RESULTS ARE ACHIEVED BY TEST ON UCID-V2 [33] CORPUS

r \ c	1	2	3	4	5	6	7	8
1	−0.0000	0.0097	0.0179	0.0427	0.0758	0.0988	0.1058	0.1111
2	0.0065	0.0249	0.0346	0.0553	0.0692	0.0907	0.0851	0.0701
3	0.0243	0.0318	0.0442	0.0673	0.0869	0.1026	0.1043	0.0863
4	0.0309	0.0488	0.0617	0.0715	0.0994	0.1278	0.0994	0.0842
5	0.0544	0.0606	0.0788	0.1016	0.1163	0.1478	0.1510	0.1410
6	0.0682	0.0621	0.0969	0.0982	0.1157	0.1301	0.1420	0.1565
7	0.0861	0.0758	0.1071	0.1013	0.1458	0.1438	0.1628	0.1547
8	0.1074	0.0968	0.1206	0.1016	0.1606	0.1516	0.1498	0.1223

A. Comparing Anti-Forensic Dithering Methods

Fig. 6 shows the SSIM value (with the original uncompressed image as the reference) comparison when tested on the classical Lena image. Note that the difference between the results in our paper and those shown in [9] is probably due to the different versions of Lena images in use and to the different parameter settings of SSIM metric. However, the trend of the curves is in accordance with each other, and the results confirm the conclusion of [9] that it is not easy to conceal the traces of JPEG compression without serious image quality loss. According to Fig. 6, the perceptual anti-forensic dither of Valenzise *et al.* [9] is able to achieve a higher SSIM value of the processed image than Stamm *et al.*'s anti-forensic dither [5]. However, the proposed DCT histogram smoothing method outperforms both of them [5], [9] in terms of SSIM value. Fig. 5 shows the close-up images of Lena. In the relatively smooth area of the image, *e.g.*, the shoulder and the face of Lena, we can see that less noise is introduced in the spatial domain of $\hat{\mathcal{F}}_{bq}$ compared with $\mathcal{F}_{S_q}$ and $\mathcal{F}_V$.

The average PSNR and SSIM values of large-scale test on UCID-v2 corpus for $\hat{\mathcal{F}}_{bq}$ are 35.9023 dB and 0.9871, respectively, which are noticeably higher than those of $\mathcal{F}_{S_q}$ and of $\mathcal{F}_V$ (see Table V for the figures). It demonstrates that the proposed perceptual DCT histogram smoothing can help us to achieve a higher image quality than the state-of-the-art anti-forensic dithering methods [5], [9].

In the proposed DCT histogram smoothing method, we have to solve an assignment problem for each quantization bin of each subband. As mentioned in Sec. IV-B.3, the computation cost is $O(D^3)$ when using the Hungarian algorithm [36], where D is the dimension of the assignment problem. Obviously, the problem solving might be impractical when D is too large. A possible solution is that we randomly split a single assignment problem into several smaller ones. From Fig. 6, we can see that the problem splitting barely affects the image quality of the processed image. In the large-scale test, in order to speed up the simulation, we always randomly split the assignment problem into several smaller ones of maximum dimension of 200.

With experiments carried out on UCID-v2 corpus, we compute the KL divergence value between $\mathcal{I}$ and $\mathcal{F}_{S_q}$, and that between $\mathcal{I}$ and $\hat{\mathcal{F}}_{bq}$ for all DCT subbands. The *difference* between these two KL divergence values is reported in Table III. We notice that during the dithering process, the subband where all the coefficients are quantized to 0 is left untouched in [5]. To be fair, these subbands were not counted in the comparison. From Table III, we can see that $\hat{\mathcal{F}}_{bq}$ performs consistently better than $\mathcal{F}_{S_q}$ except the DC component, with a smaller KL divergence value for all the 63 AC subbands. Similar results are obtained when compared with Valenzise *et al.*'s JPEG forgeries $\mathcal{F}_V$ [9], because the dithering model used in [9] is exactly the same as in [5] in the DCT domain. We can conclude that the proposed adaptive

TABLE IV
JPEG FORENSIC DETECTORS

K_F	the JPEG blocking artifact detector [3];
K_F^Q	the quantization table estimation based detector in [3];
K_{Luo}	the JPEG identifying detector [4];
K_{Luo}^Q	the quantization step estimation based detector in [4];
K_V	the TV-based JPEG forensic detector [24];
K_L	the calibration-based detector [8];
K_U^1, K_U^2	the JPEG blocking artifact detectors [12];
K_{Li}^{S100}	the 100-dimensional intra- and inter-block correlation feature [11] based detector [10];
K_P^{S162}	the 162-dimensional SPAM feature based detector [14].

local dithering model has a better restoration capability of the original DCT histogram than the one based on the global Laplacian model [5], [9].

B. Against JPEG Forensic Detectors

In this paper, we adopt the minimal decision error P_e, which is a commonly used measure in steganalysis [39], [40], for the performance evaluation of forgeries against forensic detectors. For its derivation, we first draw the receiver operating characteristic (ROC) curve of various forensic detectors, taking JPEG (anti-forensic) images as positive instances, and genuine, uncompressed images as negative instances. The minimal decision error P_e corresponds to the point on the ROC curve, which has the minimal number of incorrectly classified images. The sum of P_e and the best accuracy (another commonly used measure [16], [19]) is equal to 1.

The JPEG forensic detectors described in Sec. III are used for testing the forensic undetectability of different kinds of images on UCID-v2 corpus. For the sake of conciseness, we hereafter use symbols for referring to the detectors, which are summarized in Table IV. Here, we mainly name the detectors directly after the feature value name, *e.g.*, K_F, K_V, K_L, and K_U^p (parameters $p = 1$, and $p = 2$ are considered) in Eqs. (1), (2), (3), and (4), respectively. The subscript of the detector name is based on the surname of the first author of the corresponding algorithm. We use the superscript 'S' together with the dimensionality of the feature vector to indicate that K_{Li}^{S100} and K_P^{S162} are SVM-based detectors. Meanwhile, the superscript 'Q' of K_F^Q and K_{Luo}^Q shows that they estimate quantization steps.

In some high-frequency subbands of highly JPEG compressed images, all the DCT coefficients are quantized to 0 and no quantization step could be determined. Hence, for detector K_{Luo}^Q, we use the number of defined estimated quantization steps greater than 1 as the final feature value. Therefore, For K_{Luo}^Q and detectors K_F, K_{Luo}, K_V, K_L, K_U^1, K_U^2, they all output one feature value for a given test image. The detector can thereafter classify one image as uncompressed or JPEG compressed by thresholding. Note that, the range of the re-compression quality factor q in Eq. (2) is shrunk to $\{45, 46, \ldots, 100\}$, as here the JPEG images in use are compressed with quality factors randomly selected from $\{50, 51, \ldots, 95\}$. Though detector K_F^Q is analyzed as not very reliable in [38], as it may result in relatively high false positive rate (detecting never compressed images as JPEG forgeries), we still test it. Following the suggestion in [3] and [5], a given image is classified as never compressed if and only if its estimated quantization table is full of entries 1 or "undetermined". In light of the finding of [38], and still following [3] and [5], we report the rate of correctly detecting JPEG forgeries for detector K_F^Q. Concerning the other detectors, all the 1338 uncompressed UCID-v2 images together with its corresponding JPEG (anti-forensic) images are used for forensic testing.

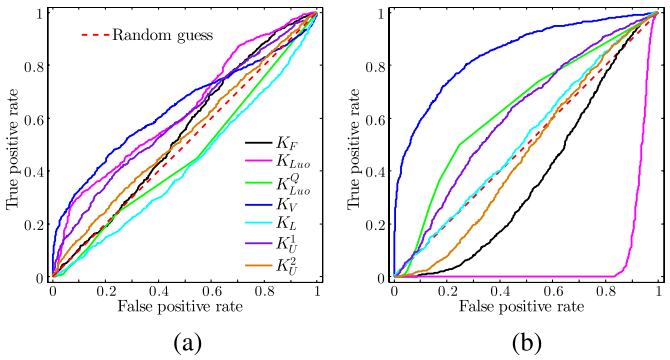


Fig. 7. ROC curves of $\mathcal{F}$ and $\mathcal{F}_{S_q S_b}$ against JPEG forensic detectors. The closer the curve is to *Random guess*, the better the detector is fooled. Note that due to lack of space in (b), its legend is removed. Please refer to (a) for different line specifiers for different JPEG forensic detectors. Results are achieved by test on UCID-v2 [33] corpus.

The minimal decision error is reported in Table V for all kinds of JPEG (anti-forensic) images against all the above described forensic detectors except K_F^Q, K_{Li}^{S100}, and K_P^{S162}. Our JPEG forgery $\mathcal{F}$ has close minimal decision errors with Stamm *et al.*'s JPEG forgery $\mathcal{F}_{S_q S_b}$. However the ROC curve of $\mathcal{F}$ is in general closer to random guess than $\mathcal{F}_{S_q S_b}$, as shown in Fig. 7. This piece of information is not always conveyed by the value of minimal decision error. Besides, the last two columns of Table V also list the average PSNR and SSIM values for the comparison of the visual quality of processed image, with the original uncompressed image as the reference.

From Table V, we can see that Stamm *et al.*'s JPEG forgery $\mathcal{F}_{S_q}$ processed by the DCT histogram smoothing can be well detected by the JPEG forensic detectors which were designed targeting at it [7], [8]. The JPEG forgery $\mathcal{F}_V$ created by the perceptual anti-forensic dithering [9] has a similar anti-forensic performance as $\mathcal{F}_{S_q}$, while achieving a higher SSIM value on average. The median filtering based deblocking [6] improves the undetectability of the JPEG forgery $\mathcal{F}_{S_q S_b}$ against forensic detectors, but with a PSNR value loss of 6.81 dB on average, compared to JPEG images.

Fig. 7-(a) shows that the proposed method succeeds in fooling all the three JPEG blocking artifact detectors, yet through the minimization of a different TV-based blocking measure in Eq. (6). Our method is also capable of fooling the advanced detector K_V. The reason might be that the TV term in Eq. (5) manages to suppress the unnatural noises utilized by this detector. Furthermore, the calibration-based detector K_L is defeated by the *decalibration* operation. Even though the quantization table estimation based detector K_F^Q is proven to be not very reliable in [38], we still test it

TABLE V

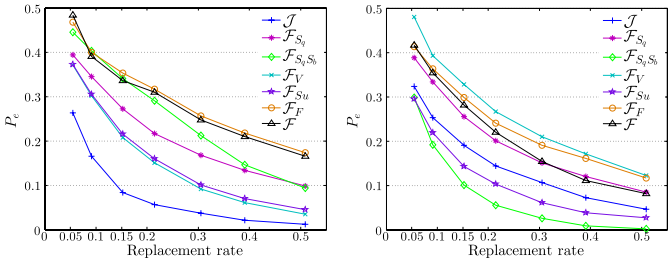
FROM THE 2ND TO THE 8TH COLUMNS, THE MINIMAL DECISION ERROR FOR DIFFERENT KINDS OF IMAGES WHEN TESTED AGAINST DIFFERENT JPEG FORENSIC DETECTORS IS LISTED; THE IMAGE QUALITY (WITH $\mathcal{I}$ AS THE REFERENCE) COMPARISON IS REPORTED IN THE LAST TWO COLUMNS. RESULTS ARE ACHIEVED BY TEST ON UCID-V2 [33] CORPUS

	K_F [3]	K_{Luo} [4]	K^Q_{Luo} [4]	K_V [7]	K_L [8]	K^1_U [12]	K^2_U [12]	PSNR	SSIM
$\mathcal{J}$	0.0082	0	0.0052	0.0108	0.0374	0.0396	0.1928	37.0076	0.9920
$\mathcal{F}_{S_q}$ [5]	0.1353	0.4645	0.2829	0.0172	0.0239	0.0848	0.1263	33.2538	0.9748
$\mathcal{F}_{S_qS_b}$ [6]	0.4865	0.4959	0.3737	0.2321	0.4712	0.4028	0.4753	30.2007	0.9480
$\mathcal{F}_V$ [9]	0.0572	0.4656	0.2313	0.0575	0.0235	0.0385	0.1147	33.1378	0.9796
$\mathcal{F}_{Su}$ [13]	0.1622	0.0807	0.0889	0.4996	0.0822	0.3393	0.4985	31.4096	0.9710
$\mathcal{F}_F$ [12]	0.3584	0.4129	0.4496	0.3632	0.5000	0.3655	0.4301	35.4047	0.9843
$\mathcal{F}$	0.4477	0.3972	0.4996	0.3756	0.5000	0.4208	0.4701	35.9019	0.9866

on our JPEG forgeries $\mathcal{F}$, 93.80% of which can be passed off as never JPEG compressed (making the rate of correctly detecting forgeries be 6.20%). Our method successfully fools existing detectors, at the cost of a slightly lower visual quality than the JPEG compressed image: 1.11 dB of PSNR loss and 0.0054 of SSIM value loss on average. Compared to Stamm *et al.*'s JPEG forgery $\mathcal{F}_{S_qS_b}$ [5], [6], our method achieves a better tradeoff between the forensic undetectability and the visual quality of processed images: the average PSNR has been improved by 5.70 dB and 0.0386 of SSIM gain has been achieved; meanwhile our method achieves a better overall forensic undetectability.

Concerning the SVM-based detectors K^{S100}_{Li}, and K^{S162}_{P} which are initially designed for steganalysis, if the training is performed on the JPEG forgeries, all kinds of current JPEG anti-forensic images fail to be undetectable. These detectors can keep a very high detection accuracy, which leads to the minimal decision error lower than 0.1. The SVM-based detectors are built based on the assumption that the forensic investigator has the knowledge of the anti-forensic method and is able to create a large amount of forgeries for training the detector. As pointed out in [2], it is a challenging task to develop anti-forensic methods capable of resisting machine learning based detectors. Nevertheless, the performance is understandable, as we have to modify a large amount of pixel values/DCT coefficients in the image to conceal JPEG footprints. In the view of image steganalysis, the modification rate (bits per pixel, bpp, or bits per non-zero DCT coefficient, bpnc) is huge.

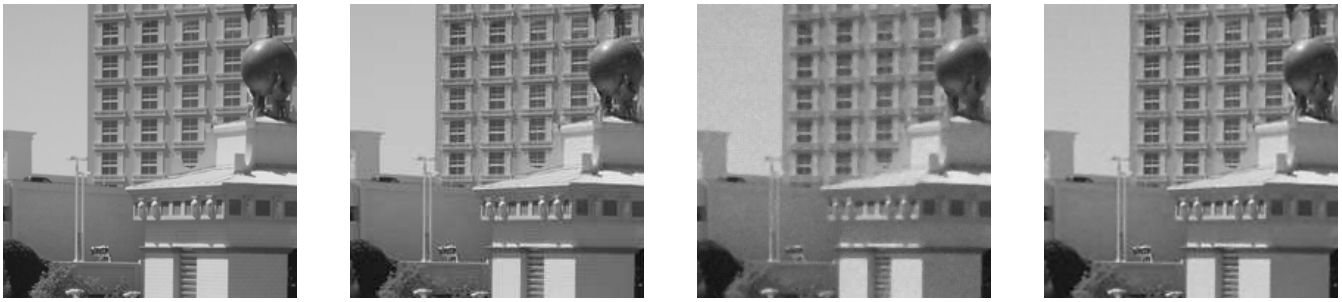
In one of the most recent steganography work [40], the undetectability of the stego images becomes poor when the payload reaches 0.5 bpp. Using a similar experimental setup to steganalytic testing, for a given uncompressed image, the center part of the image is replaced by the JPEG (anti-forensic) image with a replacement rate ranging from around 0.05 to about 0.50 to create the forgery. The replacement rate in forensic testing can be considered as a counterpart of bpp/bpnc in steganalysis. The processed images together with the uncompressed images are put together for forensic testing. The training is carried out using LIBSVM [41] with a Gaussian kernel. The parameters of the SVM classifier is obtained using the five-fold cross validation with the multiplicative grid suggested in [14]. The SVM-based classifiers trained on uncompressed images from UCIDTrain and their corresponding JPEG (anti-forensic) images are then used for forensic testing on images created from UCIDTest. The minimal decision error as a function of the replacement rate for different kinds of images, is shown in Fig. 8. We can see that our JPEG forgery $\mathcal{F}$ does not perform the best when tested by K^{S100}_{Li} and K^{S162}_{P}. However, unlike $\mathcal{F}_V$ or $\mathcal{F}_{S_qS_b}$, the performance of $\mathcal{F}$ is quite stable against the two detectors. Meanwhile, $\mathcal{F}_F$ created using our previous work [12] outperforms $\mathcal{F}$. This is understandable, as we explicitly smooth DCT histograms for creating $\mathcal{F}$, which may lead to more changes in image statistics. When the replacement rate is about 0.10, the performance of our JPEG forgery is quite satisfactory, with a relatively high minimal decision error of the SVM-based detectors. In this case, we can safely replace a 112×160 block in a 384×512 UCID-v2 image. This is enough for many forgery creation scenarios, *e.g.*, replacing the head of one person in the picture. We remain reserved on whether the proposed JPEG anti-forensic method is able to disguise a whole JPEG image as uncompressed, as it can still be well detected using machine learning methods. However, it is still highly applicable in various JPEG anti-forensic scenarios, *e.g.*, image splicing, and disguising double JPEG compression artifacts (see Sec. VI).



(a) (b)

Fig. 8. The minimal decision error as a function of image replacement rate for different kinds of images, when tested using the SVM-based detectors. Results are achieved by test on UCIDTest. (a) K^{S100}_{Li}. (b) K^{S162}_{P}.

Fig. 9 shows the JPEG anti-forensic images created from an example JPEG image with quality factor 50. As expected, Fig. 9-(d) processed using the proposed JPEG anti-forensic method better preserves the image details and the edges than -(c). It can also be observed that even after the deblocking operation [6], the spatial-domain noise introduced by the dithering signal [5] can still be noticed at the smooth areas of the image in -(c). Some example DCT histograms of the JPEG forgery created by our method are shown in the Supplementary Material: no noticeable artifacts appear in the histogram.



(a) (b) (c) (d)

Fig. 9. Example results (close-up images) of $\mathcal{F}$ compared with $\mathcal{I}$ (an uncompressed image from UCID-v2 [33] corpus), $\mathcal{J}$, and $\mathcal{F}_{S_q S_b}$, where $\mathcal{J}$ is compressed with quality factor 50. Our JPEG forgery $\mathcal{F}$ has a better image quality compared with $\mathcal{F}_{S_q S_b}$ [5], [6]. (a) $\mathcal{I}$. (b) $\mathcal{J}$. (c) $\mathcal{F}_{S_q S_b}$. (d) $\mathcal{F}$.

TABLE VI
COMPARISON OF THE AVERAGE TIME TAKEN TO CREATE DIFFERENT KINDS OF JPEG FORGERIES. RESULTS ARE ACHIEVED ON UCID92

	$\mathcal{F}_{S_q}$	$\mathcal{F}_{S_q S_b}$	$\mathcal{F}_V$	$\mathcal{F}_{Su}$	$\mathcal{F}_F$	$\mathcal{F}$
in second(s)	0.0866	0.1195	0.5415	0.0248	15.7191	218.6710

C. Computation Cost

In our experiments, the creation of JPEG forgeries is performed using MATLAB R2013a, on a PC with 4G RAM and a 2.93GHz CPU. Table VI records the average time taken for different JPEG forgery generations on UCID92. We can see that the proposed JPEG anti-forensic method requires around 3.6 minutes to create a JPEG forgery on average, using the unoptimized MATLAB code. The bottleneck of the computation cost lies in the perceptual DCT histogram smoothing, as the computation cost is $O(D^3)$ using the Hungarian algorithm. However, the reduction of the dimensionality of the assignment problem discussed in Sec. V-A can effectively lower the computation cost. Apparently, the proposed method is more complex and more computationally demanding compared to other state-of-the-art JPEG anti-forensic methods. However, the benefit is a better tradeoff between the forensic undetectability and the image quality as shown in Sec. V-B. Moreover, in practice, usually the forger does not need to create a large amount of forgeries. It is therefore acceptable to take around 4 minutes to generate a JPEG forgery.

VI. HIDING TRACES OF DJPG COMPRESSION ARTIFACTS

In the this section and the Supplementary Material[4], we show how our JPEG anti-forensic method can be used to deceive three state-of-the-art image forensic algorithms [16], [17], [19] that detect double JPEG compression artifacts.

In the following, single JPEG compressed images are only JPEG compressed with quality factor QF_2, while double JPEG compressed images are firstly JPEG compressed with quality factor QF_1 and JPEG compressed again with quality factor QF_2. Image cropping or content modification may occur between the two compressions. For anti-forensic testing, the JPEG anti-forensic methods in [5], [6], [9], [12], [13] and the proposed JPEG anti-forensic method are applied on the JPEG image after the first compression with QF_1. Then the JPEG forgery is either unchanged, or cropped by a random grid shift, or partly altered, according to different testing scenarios (see Sec. VI-A and the Supplementary Material). Finally the forgery is JPEG compressed again with QF_2 to create anti-forensic double JPEG compressed image.

We still conduct large-scale tests on UCID-v2 [33], while following the experimental settings in the original papers of double JPEG detectors [16], [17], [19]. In order to avoid tedious repetition later, we hereby explain the naming rule of the image datasets. The name of the dataset is written in bold letters, the end of which (*i.e.*, '-***R***') indicates which kind of (or no) JPEG anti-forensics is applied to the single JPEG compressed image with QF_1 before further processing. It could the followings:

- '-**T**': no JPEG anti-forensics is applied;
- '-**S**': Stamm *et al.*'s dithering based JPEG anti-forensics [5] is applied;
- '-**SS**': Stamm *et al.*'s dithering [5] and median filtering based JPEG anti-forensics [6] is applied;
- '-**V**': Valenzise *et al.*'s perceptual dithering based JPEG anti-forensics [9] is applied;
- '-**Su**': Sutthiwan and Shi's SAZ attack [13] is applied;
- '-$\mathbf{F}_0$': our previously proposed TV-based anti-forensics [12] is applied;
- '-**F**': the proposed JPEG anti-forensics is applied.

A. Hiding Traces of Non-Aligned Double JPEG Compression

Bianchi and Piva [19] analyzed the statistical change in the DCT coefficients of the DC component, after a JPEG image is compressed again with a non-aligned 8×8 grid. A simple but powerful threshold detector is constructed [19], which is based on measuring the non-uniformity of a suitably defined integer periodicity map of the DC DCT coefficients.

We compress each UCIDTest image with QF_1, then the JPEG image is cropped with a random shift $(i, j) \neq (0, 0)$, with $0 \leq i, j \leq 7$. At last the cropped image is JPEG compressed again with QF_2 to create the NA-DJPG compressed image. Meanwhile each UCIDTest image is JPEG compressed once with QF_2 for creating the single JPEG compressed image. Anti-forensic operation may occur after the first JPEG compression with QF_1. For forensic testing, we create 7 datasets whose names follow the

[4]Due to the space limit here, the experimental results against the A-DJPG detector [17] and the forgery localization detector [16] are presented in the Supplementary Material.

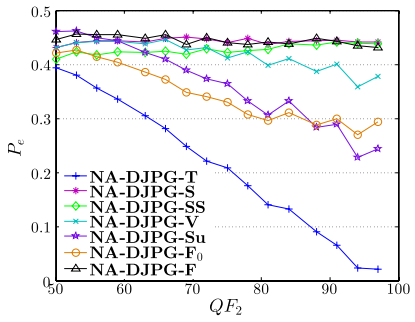


Fig. 10. Average minimal decision error of the NA-DJPG compression detector [19] as a function of QF_2, when tested on **NA-DJPG-*R***, created from UCIDTest.

TABLE VII
IMAGE QUALITY (WITH THE UNCOMPRESSED IMAGE AS THE REFERENCE) COMPARISON OF (ANTI-FORENSIC) DOUBLE JPEG COMPRESSED IMAGES CREATED FROM UCIDTEST FOR DIFFERENT DATASETS **NA-DJPG-*R***

	-T	**-S**	**-SS**	**-V**	**-Su**	**-F$_0$**	**-F**
PSNR	34.4958	32.6551	29.9884	32.3374	30.6671	33.7166	33.9723
SSIM	0.9332	0.8634	0.8450	0.8860	0.8889	0.9182	0.9235

pattern **NA-DJPG-*R***. Here, as suggested in [19], $QF_1 \in \{50, 53, 56, 59, 63, 66, 69, 72, 75, 78, 81, 84, 88, 91, 94\}$ and $QF_2 \in \{50, 53, 56, 59, 63, 66, 69, 72, 75, 78, 81, 84, 88, 91, 94, 97\}$, yielding in total $15 \times 16 = 240$ different quality factor combinations for creating NA-DJPG compressed images. Therefore, we have $15 \times 16 \times 669 + 16 \times 669 = 171264$ images for each dataset.

For each quality factor pair (QF_1, QF_2), the single JPEG compressed images together with their corresponding (anti-forensic) double JPEG compressed images are tested using the NA-DJPG detector [19]. Then the minimal decision error P_e can be computed for different kinds of images. Fig. 10 shows the average minimal decision error over quality factor QF_1, under a fixed value of QF_2. We can see that $\mathcal{F}_{S_q}$ [5], $\mathcal{F}_{S_q S_b}$ [6], both of which have DCT histograms explicitly smoothed, keep a good forensic undetectability against this NA-DJPG detector. Although our previous JPEG anti-forensic method [12] can successfully fool many existing JPEG forensic detectors (as shown in [12]), the gaps in the DCT domain are not well filled, which might be exposed by the NA-DJPG compression detector [19]. Similarly, anti-forensic double JPEG compressed images created from $\mathcal{F}_{Su}$ [13] can also be detected, to some extent, by the NA-DJPG compression detector [19], especially when QF_2 is high. By contrast, with the application of the proposed JPEG anti-forensic method, the minimal decision error of the NA-DJPG detector [19] is successfully kept close to 0.5 (random guess). This proves the necessity of an explicit DCT histogram smoothing for JPEG anti-forensics and the effectiveness of the proposed perceptual histogram smoothing method, because no integer periodicity can be detected by the NA-DJPG detector in the DCT histogram of our anti-forensic double JPEG compressed images. Moreover, as reported in Table VII, our forgeries have the highest visual quality (with the uncompressed image as the reference), among different anti-forensic double JPEG compressed images.

As shown in the Supplementary Material, the proposed JPEG anti-forensic method can also successfully fool the DJPG detectors in [16], [17], while achieving the highest visual quality of the processed image among all the six kinds of forgeries.

VII. CONCLUDING REMARKS

In this paper, we propose a novel JPEG anti-forensic method, which is able to fool JPEG compression detectors and achieve a better image visual quality than state-of-the-art methods. The proposed four-step JPEG anti-forensic method is indeed heuristic, but effective. It is an interesting but very difficult problem to design a single-step attack to remove JPEG artifacts, as existing detectors work in both spatial and DCT domains. Furthermore, we have to consider the visual quality of the processed image. As known, SSIM is non-convex, which makes it hard to optimize. In order to drag the processed image out from the detection regions of *multiple* detectors working in *different domains*, and at the same time to keep a *high image quality* under the evaluation of *both PSNR and SSIM metrics*, in each of the four steps we consider a different optimization problem. Moreover, similar to Stamm *et al.*'s JPEG forgery creation process [5], [6], the JPEG artifacts in the spatial domain and the DCT domain are considered alternatively. With a local modification to the image considering multiple metrics, we are able to create JPEG forgeries with a better tradeoff between the forensic undetectability and the image quality.

Future research shall be devoted to the design of an optimal attack to the JPEG image considering multiple detectors and the non-convex SSIM metric. We may get inspirations from existing work on optimal attack to a single, histogram-based forensic detector [18], [37]. We would like to further study the image statistics in the DCT domain for a better histogram restoration, and to compare our adaptive local dithering model with the recently proposed calibration-based non-parametric DCT quantization noise estimation method [38]. Finally, although experimental results well demonstrate that the proposed JPEG anti-forensic method has an improved tradeoff between the forensic undetectability and the image visual quality, the method is rather designed to fool targeted, well-defined JPEG forensic detectors. In the future, we plan to investigate the design and implementation of a universal framework for general-purpose anti-forensics.

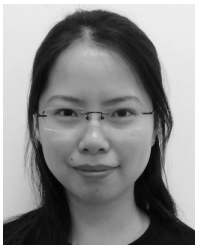
ACKNOWLEDGMENT

The authors would like to thank Dr. X. He for her insightful discussion about the total variation, and the anonymous reviewers for their valuable comments which have helped improve the quality of the paper.

REFERENCES

[1] H. Farid, "A survey of image forgery detection," *IEEE Signal Process. Mag.*, vol. 2, no. 26, pp. 16–25, Apr. 2009.
[2] 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, 2013, pp. 327–366.
[3] Z. Fan and R. L. De Queiroz, "Identification of bitmap compression history: JPEG detection and quantizer estimation," *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 230–235, Feb. 2003.

[4] W. Luo, J. Huang, and G. Qiu, "JPEG error analysis and its applications to digital image forensics," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 3, pp. 480–491, Sep. 2010.
[5] M. Stamm, S. Tjoa, W. S. Lin, and K. J. R. Liu, "Anti-forensics of JPEG compression," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2010, pp. 1694–1697.
[6] M. Stamm, S. Tjoa, W. S. Lin, and K. J. R. Liu, "Undetectable image tampering through JPEG compression anti-forensics," in *Proc. 17th IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 2109–2112.
[7] G. Valenzise, V. Nobile, M. Tagliasacchi, and S. Tubaro, "Countering JPEG anti-forensics," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 1949–1952.
[8] S. Lai and R. Böhme, "Countering counter-forensics: The case of JPEG compression," in *Proc. Int. Conf. Inf. Hiding*, 2011, pp. 285–298.
[9] G. Valenzise, M. Tagliasacchi, and S. Tubaro, "The cost of JPEG compression anti-forensics," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2011, pp. 1884–1887.
[10] H. Li, W. Luo, and J. Huang, "Countering anti-JPEG compression forensics," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2012, pp. 241–244.
[11] C. Chen and Y. Q. Shi, "JPEG image steganalysis utilizing both intrablock and interblock correlations," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2008, pp. 3029–3032.
[12] W. Fan, K. Wang, F. Cayre, and Z. Xiong, "A variational approach to JPEG anti-forensics," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2013, pp. 3058–3062.
[13] P. Sutthiwan and Y. Q. Shi, "Anti-forensics of double JPEG compression detection," in *Proc. Int. Workshop Digital Forensics Watermarking*, 2011, pp. 411–424.
[14] T. Pevny, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 215–224, Jun. 2010.
[15] E. Kee, M. K. Johnson, and H. Farid, "Digital image authentication from JPEG headers," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 1066–1075, Sep. 2011.
[16] T. Bianchi and A. Piva, "Image forgery localization via block-grained analysis of JPEG artifacts," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 1003–1017, Jun. 2012.
[17] T. Pevny and J. Fridrich, "Detection of double-compression in JPEG images for applications in steganography," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 2, pp. 247–258, Jun. 2008.
[18] P. Comesaña-Alfaro and F. Pérez-González, "Optimal counterforensics for histogram-based forensics," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Jun. 2013, pp. 3048–3052.
[19] T. Bianchi and A. Piva, "Detection of nonaligned double JPEG compression based on integer periodicity maps," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 842–848, Apr. 2012.
[20] B. Pennebaker and L. Mitchell, *JPEG Still Image Data Compression Standard*. New York, NY, USA: Van Nostrand Reinhold, 1993.
[21] P. Sallee. (2003). *MATLAB JPEG Toolbox* [Online]. Available: http://dde.binghamton.edu/download/feature_extractors/
[22] E. Y. Lam and J. W. Goodman, "A mathematical analysis of the DCT coefficient distributions for images," *IEEE Trans. Image Process.*, vol. 9, no. 10, pp. 1661–1666, Oct. 2000.
[23] Z. Wei and K. N. Ngan, "Spatio-temporal just noticeable distortion profile for grey scale image/video in DCT domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 3, pp. 337–346, Mar. 2009.
[24] G. Valenzise, M. Tagliasacchi, and S. Tubaro, "Revealing the traces of JPEG compression anti-forensics," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 2, pp. 335–349, Feb. 2013.
[25] H. Farid, "Exposing digital forgeries from JPEG ghosts," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 4, pp. 154–160, Mar. 2009.
[26] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D, Nonlinear Phenomena*, vol. 60, nos. 1–4, pp. 259–268, 1992.
[27] J. Fridrich, M. Goljan, and D. Hogea, "Steganalysis of JPEG images: Breaking the F5 algorithm," in *Proc. Int. Workshop Inf. Hiding*, 2003, pp. 310–323.
[28] C. Ullerich and A. Westfeld, "Weaknesses of MB2," in *Proc. Int. Workshop Digital Watermarking*, 2008, pp. 127–142.
[29] 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.
[30] F. Alter, S. Durand, and J. Froment, "Adapted total variation for artifact free decompression of JPEG images," *J. Math. Imag. Vis.*, vol. 23, no. 2, pp. 199–211, 2005.
[31] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[32] D. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA, USA: Athena Scientific, 1999.
[33] G. Schaefer and M. Stich, "UCID—An uncompressed colour image database," *Proc. SPIE*, pp. 472–480, Mar. 2004.
[34] J. R. Price and M. Rabbani, "Biased reconstruction for JPEG decoding," *IEEE Signal Process. Lett.*, vol. 6, no. 12, pp. 297–299, Dec. 1999.
[35] M. A. Robertson and R. L. Stevenson, "DCT quantization noise in compressed images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 27–38, Jan. 2005.
[36] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, 1955.
[37] M. Barni, M. Fontani, and B. Tondi, "A universal technique to hide traces of histogram-based image manipulations," in *Proc. ACM Workshop Multimedia Security*, 2012, pp. 97–104.
[38] W. Fan, K. Wang, F. Cayre, and Z. Xiong, "JPEG anti-forensics using non-parametric DCT quantization noise estimation and natural image statistics," in *Proc. ACM Int. Workshop Inf. Hiding Multimedia Security*, 2013, pp. 117–122.
[39] T. Pevny, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Proc. Int. Workshop Inf. Hiding*, 2010, pp. 161–177.
[40] V. Holub and J. Fridrich, "Digital image steganography using universal distortion," in *Proc. ACM Int. Workshop Inf. Hiding Multimedia Security*, 2013, pp. 59–68.
[41] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, Jan. 2011.



Wei Fan received the B.S. degree in computer science and engineering from Beihang University, Beijing, China, in 2008, where she is currently pursuing the Ph.D. degree in computer application technology with the School of Computer Science and Engineering. She is also pursuing the joint Ph.D. degree in signal, image, parole, télécoms (SIPT) with GIPSA-Lab, Grenoble INP, Grenoble, France. Her current research interests mainly include multimedia security.

Kai Wang received the Ph.D. degree in computer science from the University of Lyon, Lyon, France, in 2009. Following a 10-month post-doctoral position at Inria Nancy, he joined the GIPSA-Lab, Grenoble INP, Grenoble, France, in 2011, as a full-time CNRS Researcher. His current research interests include multimedia security and surface analysis.

François Cayre received the Ph.D. degree from the Telecom ParisTech, Paris, France, and the Université Catholique de Louvain, Louvain-la-Neuve, Belgium, in 2003. He was a Post-Doctoral Fellow with the Institut National de Recherche en Informatique et Automatique, Rennes, France, until 2005, when he joined Grenoble INP as an Assistant Professor. His current research interests include watermarking security and multimedia security at large.

Zhang Xiong is a Professor with the School of Computer Science and Engineering, Beihang University, Beijing, China, and the Director of the Advanced Computer Application Research Engineering Center with the National Educational Ministry of China. During his academic career with Beihang University for more than 20 years, he has authored over 100 referred papers in international journals and conference proceedings, and received the National Science and Technology Progress Award. His research interests and publications span from smart cities, data vitalization, computer vision, and intelligent transportation systems.