Revealing the Trace of High-Quality JPEG Compression Through Quantization Noise Analysis

Bin Li, *Member, IEEE*, Tian-Tsong Ng, Xiaolong Li, Shunquan Tan, *Member, IEEE*, and Jiwu Huang, *Senior Member, IEEE*

***Abstract*— To identify whether an image has been JPEG compressed is an important issue in forensic practice. The state-of-the-art methods fail to identify high-quality compressed images, which are common on the Internet. In this paper, we provide a novel quantization noise-based solution to reveal the traces of JPEG compression. Based on the analysis of noises in multiple-cycle JPEG compression, we define a quantity called forward quantization noise. We analytically derive that a decompressed JPEG image has a lower variance of forward quantization noise than its uncompressed counterpart. With the conclusion, we develop a simple yet very effective detection algorithm to identify decompressed JPEG images. We show that our method outperforms the state-of-the-art methods by a large margin especially for high-quality compressed images through extensive experiments on various sources of images. We also demonstrate that the proposed method is robust to small image size and chroma subsampling. The proposed algorithm can be applied in some practical applications, such as Internet image classification and forgery detection.**

***Index Terms*— Discrete cosine transform (DCT), compression identification, forward quantization noise, forgery detection.**

Manuscript received June 12, 2014; revised September 26, 2014 and December 14, 2014; accepted December 23, 2014. Date of publication January 6, 2015; date of current version February 2, 2015. This work was supported in part by the 973 Program under Grant 2011CB302204, in part by the National Natural Science Foundation of China under Grant 61103174, Grant U1135001, Grant 61332012, and Grant 61402295, in part by the Guangdong Natural Science Foundation, in part by the Foundation for Distinguished Young Talents in Higher Education of Guangdong under Grant 2012LYM_0117, and in part by the Fundamental Research Program of Shenzhen City under Grant JCYJ20140418182819173 and Grant GJHS20120621142753525. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Chiou-Ting Hsu.

B. Li is with the College of Information Engineering, Shenzhen University, Shenzhen 518060, China, and also with the Shenzhen Key Laboratory of Advanced Communications and Information Processing, Shenzhen 518060, China (e-mail: libin@szu.edu.cn).

T.-T. Ng is with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore 138632 (e-mail: ttng@i2r.a-star.edu.sg).

X. Li is with the Institute of Computer Science and Technology, Peking University, Beijing 100871, China (e-mail: lixiaolong@pku.edu.cn).

S. Tan is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, 518060 China, and also with the Shenzhen Key Laboratory of Media Security, Shenzhen, 518060, China (e-mail: tansq@szu.edu.cn).

J. Huang is with the College of Information Engineering, Shenzhen University, Shenzhen 518060, China, and also with the Shenzhen Key Laboratory of Media Security, Shenzhen, 518060, China (e-mail: jwhuang@szu.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.2015.2389148

I. Introduction

THE popularization of imaging components equipped in personal portable devices, together with the rapid development of the high-speed Internet, makes digital images become an important media for communications. Various types of image compression standards, including lossy and lossless, coexist due to different kinds of requirements on image visual quality, storage, and transmission. Among them, JPEG is a very popular lossy compression format.

Knowledge about the JPEG compression history of images from unknown sources is of important interest to image forensics experts, whose aim is to trace the processing history of an image and detect possible forgeries [1], [2]. There are some reported works on identifying whether an image is uncompressed or has been compressed previously [3], [4], whether an image has been compressed once or twice [5]–[10], whether an JPEG image has been compressed again with a shifted JPEG grid position [11]–[15], and on estimating the JPEG quantization table [16] or quantization steps [4], [17]–[21].

In this paper, we focus on the problem of identifying whether an image currently in uncompressed form is truly uncompressed or has been previously JPEG compressed. Being able to identify such a historical record may help to answer some forensics questions related to the originality and the authenticity of an image, such as where is the image coming from, whether it is an original one, or whether any tampering operation has been performed [4]. For example, the solution facilitates the detection of image forgeries created by replacing a part of an image with a fragment from another image with a different compression historical record. The mismatch of historical records reveals the act of image tampering. The JPEG identification problem [3], [4] may also be the starting point for other forensics applications, such as JPEG quantization step estimation [4], [17]–[20], for that forensics experts can save time by only performing estimation on the decompressed images after filtering out the uncompressed images.

There are also some techniques, called JPEG anti-forensics [22], [23], aiming to fool the forensics detectors by concealing the traces of JPEG compression. However, as noted by [24], removing the traces of JPEG compression is not an easy task. Some targeted anti-forensics

detectors [25]–[27] are designed to detect the traces left by anti-forensics operations.

According to the results of our random crawling on three main Internet search engines (presented in Section V-A), images with high-quality JPEG compression (where most of the quantization steps are close to 1) are not rare. They are very similar to uncompressed images due to their nearly lossless nature. High-quality JPEG compressed images are possibly preferred to be used with the uncompressed images for creating forgeries. Current forensics detectors [3], [4] are not capable of detecting high-quality compressed images even in the absence of anti-forensics operations. It is an open problem to identify high-quality compressed images when they are decompressed and re-saved in an uncompressed form.

Traces of JPEG compression may be found directly in the spatial domain (image intensity domain). Quantizing the high-frequency DCT (discrete cosine transform) coefficients with a quantization table containing large quantization steps produces ringing effects when a JPEG image is decompressed. In the case of heavy compression, undesired blocky artifacts [28], [29] will become obvious. Fan *et al.* [3] computed the statistics of differences between pixel intensity within an 8×8 block and that spanning across a block boundary, and then decided whether an image had been previously JPEG compressed by using the discrepancy between the two statistics. This method is effective for detecting severe compression which produces prominent blocky artifacts. However, in the case of high-quality compression or when the image is of small size, the statistics will not be reliable, as indicated in [4].

Traces of JPEG compression may also be found in the histogram of DCT coefficients. Luo *et al.* [4] noted that JPEG compression reduces the amount of DCT coefficients with an absolute value no larger than one. There are less DCT coefficients in the range of $[-1, 1]$ after JPEG compression. A discriminative statistics based on measuring the amount of DCT coefficients in the range of $[-2, 2]$ is constructed. When the statistics of a test image exceeds a threshold, it is classified as uncompressed. Otherwise, it is identified as having been previously JPEG compressed. Although Luo *et al.*'s method is considered as the current state of the art in terms of its identification performance, it has a few shortcomings. First, the analysis only uses a portion of the DCT coefficients that are close to 0. Hence, information is not optimally utilized. Second, the method requires the quantization step to be no less than 2 to be effective. As a result, this method fails on high-quality compressed image such as those with a quantization table containing mostly quantization steps being ones. Lai and Böhme [25] built a calibrated feature based detector, which utilizes the relation between the variance of high-frequency DCT coefficients of a given image and that of a calibrated image [30]. It is based on the assumption that the obtained statistics will be small for an uncompressed image, while the statistics will become large for an image with anti-forensics operations. The detector is effective to detect anti-forensics operations and may also be directly applicable to detect decompressed images.

Built on a theoretical model on multi-cycle JPEG compression in our previous work [31], we try to reveal the high-quality compression traces in the "noise domain". In this paper, we define a quantity, called *forward quantization noise*, and develop a simple yet very effective algorithm to judge whether an image has been JPEG compressed based on the variance of forward quantization noise. The method fully utilizes the noise information from DCT coefficients; therefore, it is neither restricted to large image size nor limited by the quantization step being no less than 2. We show that our method outperforms the previous methods by a large margin for high-quality JPEG compressed images which are common on the Internet and present a challenge for identifying their compression history.

This paper is organized as follows. Section II introduces the results from a theoretical work analyzing the noise in multi-cycle JPEG compression. Based on the analysis, we show how the variance of quantization noise can be employed to detect JPEG compression in Section III. Extensive experiments are provided in Section IV, where we demonstrate the results on gray-scale images and on color images with different chroma sub-sampling factors. Various sources of images, different definitions of JPEG quality factor, and different evaluation metrics are used to enhance the reliability of the experiments. We show possible applications to Internet image classification and image forgery detection in Section V. The paper is concluded in Section VI.

II. JPEG Quantization Noise Analysis

A JPEG compression cycle consists of an encoding phase and a decoding phase [32]. In the encoding phase, irreversible information loss occurs due to *quantizing* DCT coefficients. The decoding phase is essentially the reverse of the encoding phase. An *integer rounding and truncation* operation occurs when JPEG coefficients are restored into image intensity representation. In a recent work [31], we presented a framework for analyzing multiple-cycle JPEG compression based on a complete JPEG compression model, in contrast to the simplified models [4], [33] that are commonly used. The analysis focused on information losses in JPEG compression which can be characterized by two types of noise, *i.e., quantization noise* (in DCT domain) and *rounding noise* (in spatial domain). The *truncation error* is ignored in the model due to its fairly low impact and hard-to-model nature as discussed in [4]. Distributions of the two types of noises at different compression cycles are derived. In this section, we introduce notational conventions and summarize some of the related results from the work.

A. Notations

Throughout the paper, the image pixels or DCT coefficients are always in upper case symbols, and the noises introduced during JPEG compression are using lower case symbols.

The block-DCT coefficients in 8×8 grid are numbered from 1 to 64. The first coefficient ($u = 1$) is the mean of all pixel values in an 8×8 block and is called a DC coefficient due

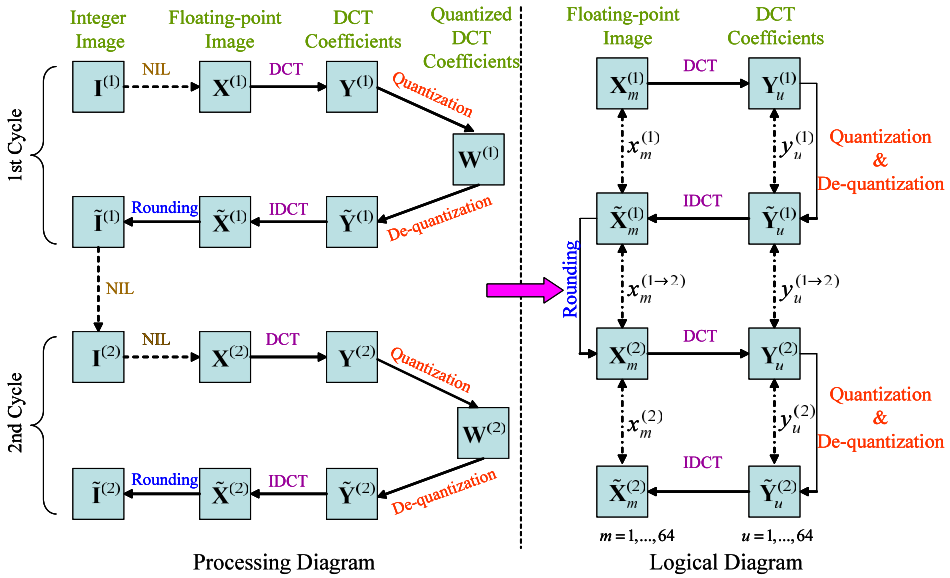


Fig. 1. Processing steps for multi-cycle JPEG compression.

to its low-pass property. The other coefficients ($u = 2, \ldots, 64$) are high-pass in nature and are called AC coefficients. The corresponding noises in DCT domain are also using the index u to indicate their locations. Similarly, the pixels in spatial domain and the corresponding noise in the same location can also be indexed from 1 to 64, and we use m to denote their indexes. We drop the frequency index u or spatial index m when there is no ambiguity.

The processing diagram for multiple-cycle JPEG compression is shown in the left part of Fig. 1, where the symbol NIL means there is no processing step. We use $X^{(k)}$ and $\tilde{X}^{(k)}$ to denote the float-point image in the JPEG encoding phase and the decoding phase, respectively, in the k-th JPEG compression cycle. We use $Y^{(k)}$ to denote the un-quantized DCT coefficients in the encoding phase, and $\tilde{Y}^{(k)}$ the de-quantized DCT coefficients in the decoding phase. The image in integer representation is denoted by $I^{(k)}$ or $\tilde{I}^{(k-1)}$, and the quantized DCT coefficients are denoted by $W^{(k)}$.

The logical diagram for multiple-cycle JPEG compression, as shown in the right part of Fig. 1, can be obtained by dropping the NIL operations from the processing diagram. In the logical diagram, we can easily define quantization noise, denoted as $y^{(k)}$, and rounding noise, denoted as $x^{(k\to k+1)}$. Besides, we define two auxiliary noise, one in spatial domain, denoted by $x^{(k)}$, and one in DCT domain, denoted by $y^{(k\to k+1)}$.

B. Quantization Noise

The information loss due to the JPEG quantization process can be referred to as quantization noise, which is defined as:

$$y = Y - \tilde{Y} = Y - \left[\frac{Y}{q}\right] q, \quad q \in \mathbb{N}, \tag{1}$$

where q is the quantization step and $[\cdot]$ represents integer rounding operation.

C. General Quantization Noise Distribution

In general, the distribution for quantization noise as defined in (1) is given by:

$$f_y(s) = \sum_{k=-\infty}^{\infty} f_Y(kq + s), \quad s \in [-\frac{q}{2}, \frac{q}{2}), \quad k \in \mathbb{Z}, \tag{2}$$

where f_y and f_Y is respectively the distribution for y and Y, and q is the quantization step. Since integer rounding is a quantization operation with $q = 1$, (2) also applies to rounding noise.

f_y is called a *quantized-Gaussian distribution* and denoted by $\mathcal{Q}^{\mathcal{N}}(\sigma^2, q)$ if Y belongs to zero-mean Gaussian distribution $\mathcal{N}(0, \sigma^2)$, where σ^2 is its variance. Its distribution function is given in (18) in Appendix B. Similarly, f_y is called a *quantized-Laplacian distribution* and denoted by $\mathcal{Q}^{\mathcal{L}}(\lambda, q)$ if Y belongs to zero-mean Laplacian distribution $\mathcal{L}(0, \lambda)$, where λ is its shape parameter and its variance equals to $2/\lambda^2$. Its distribution function is given in (16) in Appendix A.

D. Specific Quantization Noise Distribution

In [31], we found that the quantization noise of the first-round compression (given in Property 1) is different from that of the second round (given in Property 2).

Property 1: The quantization noise of the first compression cycle has the following distributions:

$$y_u^{(1)} \sim \begin{cases} \mathcal{U}(-\frac{q_1^{(1)}}{2}, \frac{q_1^{(1)}}{2}), & u = 1 \\ \mathcal{Q}^{\mathcal{L}}(\lambda_{Y_u^{(1)}}, q_u^{(1)}), & u \in \{2, 3, \ldots, 64\}, \end{cases} \tag{3}$$

where $q_u^{(1)}$ is the quantization step of the u-th frequency in the first compression cycle, and $\mathcal{U}$ represents a uniform distribution with the indicated lower and upper supports.

Property 2: For all DCT coefficient u, the second-cycle quantization noise follows the following distributions:

$$y_u^{(2)} \sim \begin{cases} \mathcal{Q}^{\mathcal{N}}(\sigma^2_{y_u^{(1\to2)}}, 1), & q_u^{(2)} = 1 \\ \mathcal{N}(0, \sigma^2_{y_u^{(1\to2)}}), & q_u^{(2)} \geq 2 \text{ and } \dfrac{q_u^{(1)}}{q_u^{(2)}} \in \mathbb{N} \\ f_y \text{ as in Equation (2)}, & \text{otherwise} \end{cases} \tag{4}$$

Note that the distribution of $y_u^{(2)}$ may depend on the variance of the auxiliary noise $y_u^{(1\to2)}$.

III. Identification of Decompressed JPEG Images Based on Quantization Noise Analysis

From above, we know that the quantization noise distributions are different in two JPEG compression cycles. In the following, we first define a quantity, call *forward quantization noise*, and show its relation to quantization noise. Then, we give the upper bound of its variance, which depends on whether the image has been compressed before. Finally, we develop a simple algorithm to differentiate decompressed JPEG images from uncompressed images.

A. Forward Quantization Noise

Given an uncompressed image, by performing the JPEG encoding phase, we can obtain its quantization noise of the first compression cycle. On the other hand, given an image that has been compressed once but stored in an uncompressed format, we can no longer retrieve the quantization noise of the first compression cycle. However, we can compute the quantization noise of the next cycle. To be unified, we call the quantization noise obtained from an image for the current available upcoming compression cycle as *forward quantization noise*.

Forward quantization noise is the subject of our analysis and it is a function of its quantization step. In this work, we study the simplest form of the forward quantization noise that corresponds to a quantization step of size one, *i.e.*,

$$z = Y - [Y], \tag{5}$$

where Y is the DCT coefficients.

For an uncompressed image, the forward quantization noise is equivalent to the first-cycle quantization noise with the quantization step being one, *i.e*, $q_u^{(1)} = 1$, $u \in \{1, \cdots, 64\}$. As stated in Property 1, we know that the forward quantization noise of the DC coefficient obtained from an uncompressed image is uniformly distributed, while those of AC coefficients are quantized-Laplacian distributed.

If a given image is compressed once, the forward quantization noise would be the quantization noise of the second compression cycle. In this case, as stated in the first condition of Property 2, since $q_u^{(2)} = 1$, $u \in \{1, \cdots, 64\}$, the forward quantization noise is quantized-Gaussian distributed.

B. Noise Variance for Uncompressed Images

For a uniform distribution $\mathcal{U}(-0.5, 0.5)$, its variance equals to $1/12$. In the following, we use $C_0 = 1/12 = 0.0833$. Given an uncompressed image, according to (3), the variance of forward quantization noise for the DC coefficients equals to C_0.

The variance of forward quantization noise for the AC coefficients is determined by the shape parameter $\lambda_{Y_u^{(1)}}$, which varies across different images and different frequency index u. However, we find that the upper bound of the variance of quantized-Laplacian distribution is related to quantization step q by the following result.

Proposition 1: The variance of a quantized-Laplacian distribution is upper-bounded by that of a uniform distribution with an identical region of support.

The proof of this proposition can be found in Appendix A. As the quantization noise distributions of AC coefficients have identical region of support with $q = 1$, their variances are upper bounded by the variance of $\mathcal{U}(-0.5, 0.5)$, which equals to C_0.

In summary, we have the following upper bound for the variance of forward quantization noise of an uncompressed image:

$$\sigma_z^2 = \sigma^2_{y^{(1)}} \leq C_0. \tag{6}$$

C. Noise Variance for Images With Prior JPEG Compression

According to the first condition of Property 2, as we use unit quantization steps, the forward quantization noise is distributed as quantized-Gaussian. We provide the following proposition to give the upper bound of the variance of the quantized-Gaussian distribution.

Proposition 2: When a zero-mean Gaussian signal $\upsilon \sim \mathcal{N}(0, \sigma^2)$ is quantized, the quantization noise, defined by $n_\upsilon = \upsilon - [\upsilon]$, is quantized-Gaussian distributed. We have the following results for the variance of the quantization noise $\sigma^2_{n_\upsilon}$:

$$\sigma^2_{n_\upsilon} \leq \begin{cases} C_0, & \text{if } \sigma^2 > C_0, \\ C_1, & \text{if } \sigma^2 \leq C_0, \\ C_2, & \text{if } \sigma^2 \leq C_1, \end{cases} \tag{7}$$

where $C_0 = 0.0833$, $C_1 = 0.0638$, and $C_2 = 0.0548$.

The derivation of the upper bounds, *i.e.,* C_0, C_1, and C_2, are obtained by firstly expressing the variance $\sigma^2_{n_\upsilon}$ using the probability density function of υ with the parameter σ^2, and then evaluating the expression numerically with the given value of σ^2. The details can be found in Appendix B.

In order to understand the property of quantization noise of the second quantization cycle, we also need to understand the variance of DCT auxiliary noise $y^{(1\to2)}$ (see (4) and Fig. 1). Its upper bound is given by the following proposition, and the proof can be found in Appendix C.

Proposition 3: The variance of the auxiliary noise $y^{(1\to2)}$ is upper bounded as follows:

$$\sigma^2_{y^{(1\to2)}} \leq \begin{cases} C_1, & \text{if } q_u^{(1)} = 1, \forall u, \\ C_0, & \text{otherwise,} \end{cases} \tag{8}$$

where $q_u^{(1)}$ is the quantization step of the first cycle.

As far as our forward quantization noise is concerned, for an image with prior JPEG compression, the forward quantization

step corresponds to $q^{(2)} = 1$. In this case, according to (4), we can further specialize Property 2 into:

Corollary 1: When $q^{(2)} = 1$ in the second quantization cycle, the corresponding quantization noise is given by:

$$y^{(2)} \sim \begin{cases} \mathcal{Q}^{\mathcal{N}}(\sigma^2_{y^{(1\to 2)}}, 1), \sigma^2_{y^{(1\to 2)}} \le C_1, & \text{if } q_u^{(1)} = 1, \forall u, \\ \mathcal{Q}^{\mathcal{N}}(\sigma^2_{y^{(1\to 2)}}, 1), \sigma^2_{y^{(1\to 2)}} \le C_0, & \text{otherwise.} \end{cases} \tag{9}$$

From Corollary 1 and Proposition 2, we have the following upper bound for the variance of forward quantization noise of an image with prior JPEG compression:

$$\sigma_z^2 = \sigma^2_{y^{(2)}} \le \begin{cases} C_2, & \text{if } q_u^{(1)} = 1, \forall u, \\ C_1, & \text{otherwise.} \end{cases} \tag{10}$$

D. Algorithm for Identifying Decompressed JPEG Images

Combining the results of (6) and (10), we have the following result about the forward quantization noise. Given a test image I, the variance of forward quantization noise z with $q = 1$ is given by:

$$\sigma_z^2 \le \begin{cases} C_0, & \text{if } I \text{ is uncompressed,} \\ C_1, & \text{if } I \text{ was compressed once.} \end{cases} \tag{11}$$

Note that the above result on noise variance is derived theoretically. The distribution of empirical data may deviates from the theoretical model because of the finite sample size. For this reason, the estimated noise variance of the empirical samples, denoted by $\hat{\sigma}_z^2$, may slightly exceed the upper bound, *i.e.*, C_1 or C_0. As observed from the distribution of $\hat{\sigma}_z^2$ for test images in our experiments in Section IV-A, the deviation decreases as the image size increases and the quality factor increases.

Since $C_1 < C_0$, we can design a reliable two-step algorithm to identify whether an image in uncompressed form has been JPEG compressed before.

1) Compute $\hat{\sigma}_z^2$ for a test image I using all block-DCT coefficients including both DC and AC coefficients.
2) Use a decision rule:

$$I = \begin{cases} \textit{uncompressed}, & \hat{\sigma}_z^2 > T, \\ \textit{decompressed}, & \hat{\sigma}_z^2 \le T, \end{cases} \tag{12}$$

where T is a predefined threshold which is in between C_1 and C_0.

The threshold T in (12) controls the trade-off between the true positive rate and the false positive rate of the detector, where we regard the decompressed images as the positive class and the uncompressed images as the negative class. We can determine the decision threshold T according to some practical requirements. To fix the detector characteristic, we can tune the threshold such that the detector has a false positive rate of 1% on a hold-out image set for specific image sizes, as given in Section IV.

As shown in (10), compared to other kinds of quantization tables, a quantization table which contains all unit quantization steps has a upper bound of the noise variance being smaller than C_1. With the decision rule in (12) where the threshold is larger than C_1, our method has a better performance against high-quality compression. It happens that this is an open problem for previous methods [3], [4], and our approach work effectively on it.

IV. Performance Evaluation

In this part, we evaluate the performance of the proposed algorithm by comparing our method with Luo *et al.*'s method [4] (referred to as Luo's method), which is better than [3] and is regarded as the current state of the art. We also use Lai and Böhme's method [25] (referred to as Lai's method) for comparison, which was targeted for countering anti-forensics purpose but may also be applicable in identifying decompressed JPEG images. The training-based method (referred to as SPAM method) [34] with the SPAM (subtractive pixel adjacency matrix) feature and the SVM (support vector machine) classifier, which was designed for steganalysis, is also included for comparison. Since it is not as flexible and time-efficient as other three methods in performing forensics-related tasks, we only use it in Section IV-A. The (Gaussian) radial basis function kernel is used in the SVM and the parameters are optimized by grid-search.

We use four different settings. Firstly, we test the methods on gray-scale images to show how the performance is on each designated compression quality. Secondly, we run test on color images to show whether the methods are robust to chroma sub-sampling. Thirdly, we conduct experiments on JPEG images from a publicly available database with random quality factors to verify the true positive rates. Finally, we conduct experiments on uncompressed images from another database to verify the false negative rates.

A. Evaluation on Gray-Scale Images With Designated Quality Factor

We conducted experiments with the following settings to validate our method on gray-scale images.

1) Image Set: Our image set is composed of 3,000 images, with 1,000 of them coming from BOSSbase ver 1.01 image database [35], 1,000 from NRCS image database [36], and 1,000 from UCID image database [37]. These publicly available image sets are a reliable source of uncompressed images. Some of them have been used in [4]. The images are first converted into gray-scale and then center-cropped to generate images of smaller sizes, *i.e.*, 256×256, 128×128, 64×64, and 32×32 pixels. The uncompressed images as well as their corresponding decompressed JPEG images are used for evaluation. In Fig. 2, we show the distribution of the pixel variance for the uncompressed images. As we expect, images of small sizes (e.g., 64×64 and 32×32) which are cropped from a large image, tend to be smooth, while images of medium sizes (e.g., 256×256 and 128×128) may contain more textures and have a larger pixel variance.

2) Evaluation Metrics: As we assume the decompressed images and uncompressed images respectively to be the positive class and the negative class, *true positive* rate and *true negative* rate respectively evaluate the percentage of correctly

TABLE I

THRESHOLD USED WHEN FALSE POSITIVE RATE IS 1% FOR THE IMAGE SET DESCRIBED IN SECTION IV-A

	256×256			128×128			64×64			32×32		
	Lai's	Luo's	Ours	Lai's	Luo's	Ours	Lai's	Luo's	Ours	Lai's	Luo's	Ours
Threshold	0.1973	0.2055	0.0822	0.3629	0.1781	0.0817	0.7080	0.1648	0.0802	1.5342	0.1414	0.0775

TABLE II

THE ACCURACY (IN %) ON IDENTIFYING GRAY-SCALE UNCOMPRESSED IMAGES AND DECOMPRESSED JPEG IMAGES

	256×256				128×128				64×64				32×32			
QF	SPAM	Lai's	Luo's	Ours	SPAM	Lai's	Luo's	Ours	SPAM	Lai's	Luo's	Ours	SPAM	Lai's	Luo's	Ours
100	62.58	49.93	50.00	**99.99**	59.42	50.00	50.00	**99.96**	57.08	49.99	50.03	**99.93**	55.33	49.83	49.87	**99.87**
99	74.50	53.90	51.03	**99.99**	68.67	50.76	50.93	**99.96**	62.75	51.09	50.49	**99.88**	59.08	51.66	51.04	**99.85**
98	86.58	65.97	97.23	**99.99**	79.67	62.78	97.03	**99.94**	68.92	63.63	96.56	**99.88**	64.92	71.09	95.55	**99.79**
95	95.67	86.53	99.93	**99.97**	90.42	84.08	99.84	**99.94**	84.50	84.65	99.71	**99.87**	75.25	91.10	99.47	**99.52**
90	98.33	94.12	99.91	**99.96**	95.67	93.08	99.80	**99.93**	91.08	94.13	99.57	**99.81**	85.33	96.53	99.33	**99.40**
85	98.92	97.59	99.90	**99.98**	97.42	96.33	99.73	**99.94**	94.67	97.33	99.50	**99.76**	90.33	98.12	99.24	**99.38**
75	99.33	99.42	99.85	**99.98**	98.42	99.15	99.68	**99.93**	96.75	99.58	99.35	**99.68**	92.58	**99.32**	98.89	99.30
50	99.67	99.96	99.74	**99.99**	99.25	99.94	99.47	**99.87**	98.42	**99.89**	99.28	99.70	97.75	**99.63**	98.71	99.19

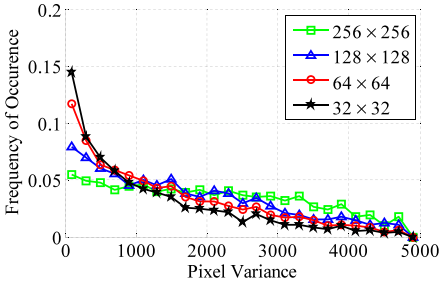


Fig. 2. Distribution of the pixel variances of uncompressed images used in Section IV-A.

identified decompressed images and that of uncompressed images. *False positive* rate evaluates the percentage of wrongly identified uncompressed images.

It is not easy to tune the parameters for the SPAM detector by non-linear SVM with a designated false positive rate. In this case, we use the metric *accuracy*, which is defined as the total amount of true positive samples and true negative samples over the total amount of test samples for each quality factor. As we report the results with accuracy, we always randomly split the images into the training set ($4/5$ of the overall images) and the testing set ($1/5$ of the overall images), and apply the threshold or the parameters, obtained on the training set with the best accuracy, to the testing set. The testing results are averaged by 5-times splitting.

As we fix a false positive rate for the whole image set, we can easily obtain a threshold respectively for Lai's, Luo's, and our method. In this case, the performance can be evaluated based on the true positive rate, the higher the better. When the false positive rate is set as 1%, the threshold of each detector is shown in Table I.

Note that for the results reported in accuracy, we may need to tune the threshold or the parameters for each quality factor. For the results reported in true positive, we only need to set the threshold according to the uncompressed images, which bring us great flexibility. We will always utilize the thresholds from Table I to other data sets (e.g., Section IV-C and IV-D) and to practical applications (e.g. Section V) for evaluation.

3) Results on Designated IJG Quality Factors: We designate IJG (Independent JPEG Group) [38] QF (quality factor) of 100, 99, 98, 95, 90, 85, 75, and 50. The results evaluated in metric *accuracy* are demonstrated in Table II. It can be observed that our method always performs the best. Luo's method starts to perform well when QF is below 98, while Lai' method and SPAM method achieve satisfactory results when QF is even lower. The identification results evaluated when the false positive rate is 1% are demonstrated in Table III. It can be observed that when QF is below 85, all three methods perform similarly. When QF is below 98, the performance difference between our method and Luo's method is marginal.

Our method shows great improvement over other methods when QF is above 98, where the quantization tables are mainly composed of small steps, *i.e.*, 1 or 2. To our knowledge, there is no other methods can distinguish uncompressed images from decompressed JPEG images with such high quality factors.

To better understand why our method can performs well on high-quality compressed images, we show the distribution of the estimated noise variance $\hat{\sigma}_z^2$ in Fig. 3 for the test images. The variances of uncompressed images are concentrated around $C_0 = 0.0833$, while that of decompressed images are concentrated around or less than $C_1 = 0.0638$. The results conform to our theoretical analysis. We demonstrate the standard deviation of the noise variance under different image size and different quality factor in Table IV. We can observe that as the image size increases or the quality factor increases, the standard deviation decreases, indicating the deviation of the empirical data from the theoretical model becomes less.

4) Results on Designated Photoshop Quality Factors: We perform experiments by using Photoshop QF of 100, 99, 98, 95, 93, 90, 85, 80, and 75. The results are shown

TABLE III

TRUE POSITIVE RATE (IN %) ON IDENTIFYING GRAY-SCALE DECOMPRESSED JPEG IMAGES WHEN FALSE POSITIVE RATE IS 1%

	256 × 256			128 × 128			64 × 64			32 × 32		
QF	Lai's	Luo's	Ours	Lai's	Luo's	Ours	Lai's	Luo's	Ours	Lai's	Luo's	Ours
100	0.87	0.20	**100.00**	0.83	0.30	**100.00**	0.80	0.37	**100.00**	0.90	0.33	**100.00**
99	0.73	0.33	**100.00**	0.63	0.30	**100.00**	0.73	0.37	**100.00**	1.43	0.40	**100.00**
98	18.13	42.37	**100.00**	15.47	15.77	**100.00**	20.87	10.10	**100.00**	39.40	5.77	**99.97**
95	69.10	**100.00**	**100.00**	64.83	**100.00**	**100.00**	69.13	**100.00**	**100.00**	83.27	**99.90**	99.73
90	88.20	**100.00**	**100.00**	86.40	99.97	**100.00**	88.80	99.93	**100.00**	93.93	**99.77**	99.70
85	95.83	**100.00**	**100.00**	93.27	99.90	**100.00**	95.17	99.83	**99.97**	97.27	**99.50**	**99.50**
75	99.30	99.97	**100.00**	98.97	99.90	**100.00**	99.17	99.73	**99.90**	99.40	99.00	**99.57**
50	99.97	99.97	**100.00**	99.90	99.77	**100.00**	99.77	99.43	**99.87**	**99.13**	98.50	**99.13**

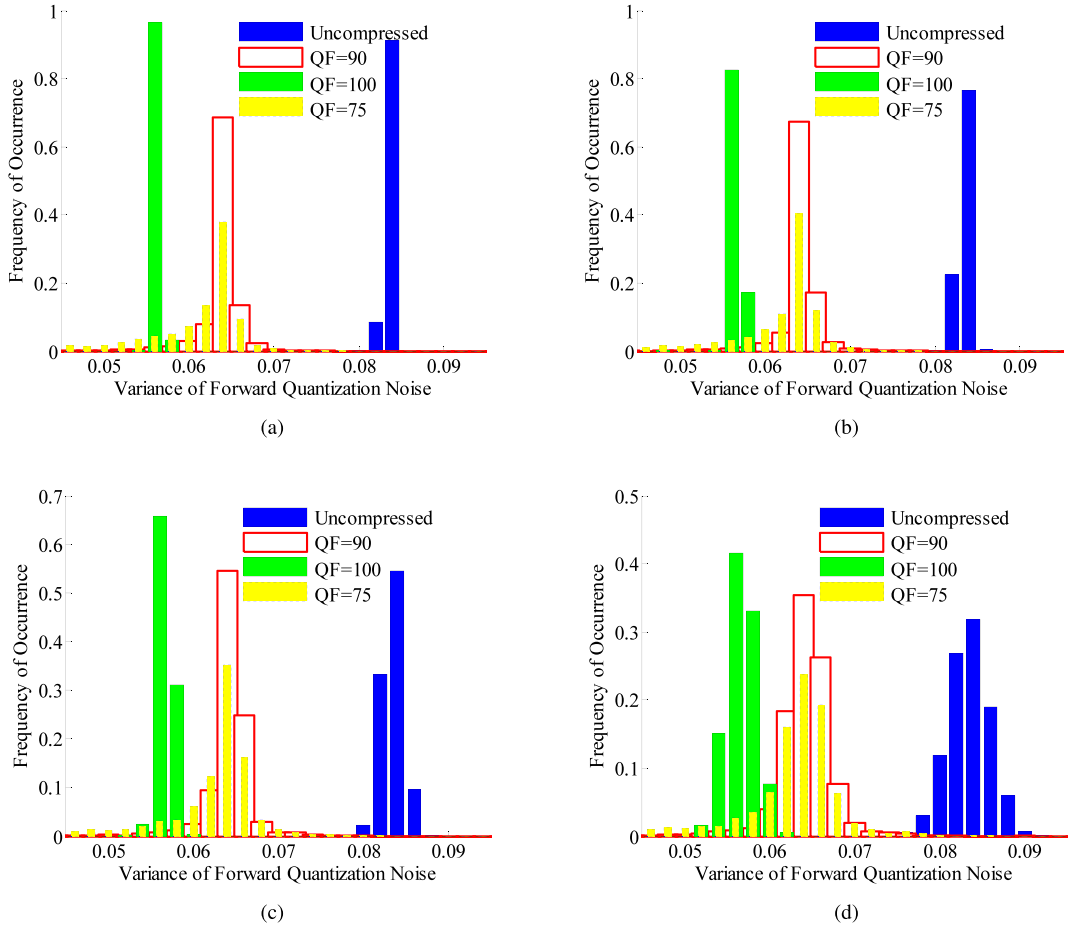


Fig. 3. Distribution of the estimated variance of forward quantization noise. (a) 256 × 256. (b) 128 × 128. (c) 64 × 64. (d) 32 × 32.

TABLE IV

STANDARD DEVIATION OF THE NOISE VARIANCE UNDER DIFFERENT IMAGE SIZE AND DIFFERENT QUALITY FACTOR

QF	256 × 256	128 × 128	64 × 64	32 × 32
100	0.0011	0.0019	0.0027	0.0034
99	0.0012	0.0021	0.0030	0.0038
98	0.0013	0.0022	0.0031	0.0039
95	0.0015	0.0023	0.0033	0.0041
90	0.0026	0.0033	0.0042	0.0050
85	0.0053	0.0059	0.0067	0.0077
75	0.0101	0.0110	0.0120	0.0134
50	0.0139	0.0153	0.0168	0.0188

in Table V. We can observe that when using a high Photoshop QF (larger than 90), our method is significantly better than Luo's method and Lai's method.

The significant performance of our method highlights the power of quantization noise analysis which reveals the great difference between uncompressed images and high-quality compressed image. Such difference turns out to be not prominent in the distribution of DCT coefficients and explains why Luo's method and Lai's method failed under such conditions.

B. Evaluation on Color Images

Since color images are pervasive in daily life, we verify the performance on color images.

1) Test Image Set: We use the same source image set as that in Section IV-A. The color images are first center-cropped to some smaller sizes, and then compressed with designated IJG QFs. During compression, we generate two types of color JPEG images. For the first type, there is no down-sampling operation on color channels. This corresponds

TABLE V

TRUE POSITIVE RATE (IN %) ON IDENTIFYING GRAY-SCALE DECOMPRESSED JPEG IMAGES BY PHOTOSHOP QUALITY FACTORS WHEN FALSE POSITIVE RATE IS 1%

	256×256			128×128			64×64			32×32		
Photoshop QF	Lai's	Luo's	Ours	Lai's	Luo's	Ours	Lai's	Luo's	Ours	Lai's	Luo's	Ours
100	6.87	0.77	**100.00**	4.10	0.73	**100.00**	4.33	0.70	**100.00**	12.13	0.50	**100.00**
99	6.77	1.17	**100.00**	4.53	0.93	**100.00**	5.53	0.77	**100.00**	14.87	0.60	**100.00**
98	15.83	1.60	**100.00**	13.87	1.03	**100.00**	18.33	0.90	**100.00**	33.40	0.60	**99.97**
95	27.53	19.77	**100.00**	24.17	6.17	**100.00**	31.40	3.93	**100.00**	53.40	2.87	**99.97**
93	39.63	66.50	**100.00**	35.20	30.63	**100.00**	41.53	21.00	**100.00**	64.20	11.40	**99.97**
90	56.73	**100.00**	**100.00**	52.37	97.93	**100.00**	58.23	92.10	**100.00**	76.73	71.97	**99.83**
85	74.27	**100.00**	**100.00**	70.57	**100.00**	**100.00**	75.00	99.97	**100.00**	88.07	**99.97**	99.87
80	85.90	**100.00**	**100.00**	82.43	**100.00**	**100.00**	83.13	99.97	**100.00**	91.60	**99.87**	99.70
75	92.70	**100.00**	**100.00**	90.30	**100.00**	**100.00**	90.53	99.97	**100.00**	94.17	**99.90**	99.77

TABLE VI

TRUE POSITIVE RATE (IN %) ON IDENTIFYING COLOR DECOMPRESSED JPEG IMAGES (WITH CHROMA SUB-SAMPLING FACTOR 4 : 4 : 4 DURING COMPRESSION) WHEN FALSE POSITIVE RATE IS 1%

	256×256			128×128			64×64			32×32		
QF	Lai's	Luo's	Ours	Lai's	Luo's	Ours	Lai's	Luo's	Ours	Lai's	Luo's	Ours
100	0.37	0.20	**100.00**	0.57	0.37	**100.00**	1.37	0.40	**100.00**	2.00	0.37	**100.00**
99	0.37	0.37	**100.00**	0.57	0.37	**100.00**	1.10	0.47	**100.00**	1.93	0.40	**99.70**
98	17.53	41.60	**100.00**	8.97	15.37	**99.97**	3.83	10.00	**99.73**	4.23	5.77	**98.87**
95	73.63	**100.00**	99.93	66.37	**99.93**	99.87	63.13	**99.73**	99.23	65.67	**99.27**	97.20
90	90.13	99.87	**100.00**	88.10	99.60	**99.97**	88.43	**99.30**	98.53	84.50	**98.17**	96.20
85	96.50	99.67	**99.90**	93.97	99.00	**99.80**	95.13	98.57	**98.77**	91.10	**97.57**	96.10
75	99.33	99.50	**99.90**	99.03	98.57	**99.80**	99.20	97.87	98.77	**98.53**	96.77	96.27
50	**99.97**	99.17	99.90	**99.90**	98.27	99.63	**99.80**	97.53	98.67	**99.67**	96.10	95.80

TABLE VII

TRUE POSITIVE RATE (IN %) ON IDENTIFYING COLOR DECOMPRESSED JPEG IMAGES (WITH CHROMA SUB-SAMPLING FACTOR 4 : 1 : 1 DURING COMPRESSION) WHEN FALSE POSITIVE RATE IS 1%

	256×256			128×128			64×64			32×32		
QF	Lai's	Luo's	Ours	Lai's	Luo's	Ours	Lai's	Luo's	Ours	Lai's	Luo's	Ours
100	0.93	0.23	**99.97**	0.87	0.40	**99.87**	0.83	0.43	**99.13**	0.93	0.43	**97.53**
99	0.77	0.33	**100.00**	0.67	0.40	**99.77**	0.77	0.47	**98.97**	1.43	0.43	**96.77**
98	18.07	37.77	**99.93**	15.20	13.07	**99.63**	20.53	8.00	**98.73**	38.83	4.73	**96.13**
95	68.87	99.83	**99.90**	64.57	99.47	**99.77**	68.90	**99.13**	98.47	82.57	**98.30**	95.67
90	88.03	99.77	**99.90**	86.20	99.20	**99.70**	88.67	**98.67**	98.53	93.67	**97.57**	95.47
85	95.80	99.60	**99.87**	93.20	98.90	**99.57**	95.10	98.37	**98.47**	97.10	**97.43**	95.03
75	99.30	99.33	**99.80**	98.97	98.43	**99.33**	**99.13**	97.47	97.90	**99.30**	96.27	95.13
50	**99.97**	98.83	99.83	**99.90**	97.83	99.23	**99.77**	97.10	97.97	**99.20**	95.37	94.80

to the chroma sub-sampling factor of 4 : 4 : 4. For the second type, we use a chroma sub-sampling factor of 4 : 1 : 1, which means the two chrominance channels are down-sampled by a factor of 2 on each dimension. These two types of color images are often found in our daily uses. We decompress the JPEG images into RGB representation for testing, and only use the luminance channel of the image as the input for each method for evaluation.

2) Evaluation Metrics and Results: We use the constant threshold giving out the false positive rate of 1% to compute the true positive rate. Since the luminance images are exactly the same as that in Section IV-A, the thresholds are the same as that in Table I.

The results on two different chroma sub-sampling types are reported in Table VI and VII, respectively. It can be observed that the trend of the performances is similar to the case of gray-scale images. The performances of Luo' method and our method may slightly decrease on color images. The reason for the performance drop is that extra noise has been introduced due to color space conversion. For our method, the variance of forward quantization noise of a decompressed image in color representation is thus larger than that in gray-scale representation. Since we use the same threshold as the gray-scale case, the true positive rate, which measures how many decompressed images have a noise variance below the threshold, decreases. The sub-sampling on the chrominance channels does not deteriorate the performance much when compared to the non-sub-sampling case, which demonstrates that applying our method only to the luminance channel is effective.

C. Evaluation on JPEG Images From a Database With Random Quality Factors

Since the decompressed JPEG images encountered in daily life are coming from different sources, and thus having been compressed with varying quality factors. We conduct the following experiment to show the performance on random quality factors.

TABLE VIII

TRUE POSITIVE RATE (IN %) ON IDENTIFYING COLOR DECOMPRESSED JPEG IMAGES FROM REWIND SYNTHESIS DATABASE

	256×256			128×128			64×64			32×32		
Image Set	Lai's	Luo's	Ours	Lai's	Luo's	Ours	Lai's	Luo's	Ours	Lai's	Luo's	Ours
Original	88.00	90.38	**99.99**	87.63	90.28	**99.96**	86.40	90.17	**99.83**	83.83	89.88	**99.21**
Class 1	87.27	89.79	**99.88**	87.24	89.69	**99.75**	85.95	89.61	**99.51**	83.71	89.33	**98.83**
Class 2	99.88	99.76	**99.92**	99.46	99.19	**99.89**	98.56	99.11	**99.78**	95.43	98.95	**99.28**
Class 3	89.43	90.44	**99.85**	88.47	90.83	**99.72**	87.45	90.75	**99.50**	85.08	90.46	**98.80**
Class 4	99.86	99.82	**99.91**	98.94	99.21	**99.85**	98.36	99.13	**99.71**	95.47	98.93	**99.17**

TABLE IX

FALSE POSITIVE RATE (IN %) ON IDENTIFYING UNCOMPRESSED IMAGES IN BOWS2 WITH THE THRESHOLDS IN TABLE I

256×256			128×128			64×64			32×32		
Lai's	Luo's	Ours	Lai's	Luo's	Ours	Lai's	Luo's	Ours	Lai's	Luo's	Ours
4.77	4.13	**2.01**	3.36	2.08	**0.98**	2.83	0.95	**0.42**	5.13	**0.36**	0.58

1) Test Image Set: To increase the amount and the diversity of images for testing, and also to test whether the thresholds of the methods heavily rely on image database, we use the test image set composed of 9,600 color JPEG images created by Fontani et al. [39], which we called REWIND SYNTHESIS database. In this database, 4,800 images are generated with IJG QFs randomly selected from the set $\{40, 50, \cdots, 100\}$. These images are referred to as "Original". The rest 4,800 images are divided into four classes (referred to as Class 1 to Class 4). Each class contains 1,200 images where aligned or non-aligned double compression operation is performed in a portion of each image. The QF in the first compression, QF_1, is randomly chosen from the set $\{40, 50, \cdots, 80\}$, and the QF in the second compression is set to $QF_2 = QF_1 + 20$. Readers can refer to [39] for details of the four classes.

Since all the images in the REWIND SYNTHESIS database are already JPEG compressed, the images are decompressed and saved in uncompressed format in our experiment to play the role of positive samples. We also divide the images of original size 1024×1024 pixels into smaller sizes. It is equivalent to the case that we have 153,600 images with size 256×256, 614,400 images with size 128×128, 2,457,600 images with size 64×64, and 9,830,400 images with size 32×32.

2) Evaluation Metrics and Results: We use the constant threshold giving out the false positive rate of 1% as that in Table I to compute the true positive rate. No matter which type (single compressed, aligned double compressed, or non-aligned double compressed) an image belongs to, the image is in the category of JPEG decompressed. A perfect detector would give a result indicating all images are positives.

The results are reported in Table VIII. Our method performs the best and it is very stable across different image types and image sizes. It can be observed that both Lai's method and Luo's method perform better on Class 2 and Class 4 than on Original, Class 1, and Class 3. This phenomenon may probably due to how the images are composed of. In Original, Class 1, and Class 3, the major part of the image is singly compressed, possibly with a high QF being close to 100 and difficult for Lai's method and Luo's method to work well.

D. Evaluation on Uncompressed Images From Another Set

Before ending the evaluation, we perform one more test to verify if the thresholds of the methods obtained in Table I is robust to other uncompressed images.

1) Test Image Set: We use the test image set composed of 10,000 uncompressed gray-scale images from BOWS2 image database [40]. The images are resized with the nearest neighbor algorithm to generate images of smaller sizes, *i.e.*, 256×256, 128×128, 64×64, and 32×32 pixels.

2) Evaluation Metrics and Results: We apply the thresholds in Table I, and evaluate the false positive rate, the lower the better. It can be observed from Table IX that the false positive rate of our method is the lowest when the image size is no smaller than 64×64, and it is close to Luo's method when image size is 32×32. The false positive rate of Lai's method is always larger than 1%, indicating the threshold used in Lai's method may highly depend on the image set, and may not be as stable as other two methods. In fact, when we compare the scale of the threshold (the ratio between the largest threshold and the smallest threshold) across different image sizes in Table I, both Luo's method ($0.2055/0.1414 = 1.4533$) and our method ($0.0822/0.0755 = 1.0887$) have a smaller scale than Lai's method ($1.5342/0.1973 = 7.7760$), which implies that these two methods may be more adaptable than Lai's method.

V. PRACTICAL APPLICATIONS

In the previous section, we have reported the performance of different methods on identifying decompressed images in designated image sizes. In practical scenarios, the methods may be applied to images with arbitrary sizes and it is infeasible to give a threshold for each individual image size. This raises a question: how to apply the methods in practical applications? In this section, we address the issue in two applications.

A. Internet Image Classification

The first application of our JPEG identification method is Internet image classification. Internet search engines cur-

rently allow users to search by content type, but not by compression history. There may be some graphic designers who wish to differentiate good-quality decompressed images from uncompressed images in a set of images returned by Internet search engines. In this case, searching images by compression history is important. In this section, we show the feasibility of such an application.

1) Image Classification Algorithm: We first convert color images into gray-scale images. Then we divide each image into non-overlapping *macro-blocks* of size $B \times B$ (e.g., $B = 128$, 64, or 32). If the dimension of the image is not exactly the multiple times of B, the last a few rows or columns are removed from testing. Next, we perform JPEG identification on each macro-block. We can use the threshold as given in Table I for each macro-block size. For a test image I, suppose it contains a total number of $N^{(B)}$ macro-blocks, and assume a number of $D^{(B)}$ macro-blocks are identified as decompressed. We use a measuring quantity, called *block hit* (BT), to assess the proportion of macro-blocks being identified, *i.e.*,

$$BT^{(B)} = \frac{D^{(B)}}{N^{(B)}}. \tag{13}$$

Ideally, $BT^{(B)}$ should be close to 1 for a decompressed image and be close to 0 for an uncompressed image. However, the results in the previous section show that none of the three methods result in 100% true positive rate and 100% true negative rate. There may be some macro-blocks identified as uncompressed in a decompressed image, and vice versa. Therefore, we make a decision rule based on:

$$I = \begin{cases} \textit{uncompressed}, & BT^{(B)} < R, \\ \textit{decompressed}, & BT^{(B)} \geq R, \end{cases} \tag{14}$$

where $R \in [0, 1]$ is a threshold controlling the classification accuracy, which will be discussed later.

2) Test Image Set: We have downloaded 15,000 color JPEG images from Internet by the main search engines, where 5,000 are from Google (http://images.google.com.hk/), 5,000 from Baidu (http://image.baidu.com/), and 5,000 from Microsoft Bing (http://cn.bing.com/images). We restrict the width or height of each image to be no smaller than 128. These images are decompressed and served as the ground-truth JPEG decompressed images. The image contents cover a wide range of semantics, including peoples, animals, buildings, landscapes, daily used goods, cartoons, logos, advertisements and so on. The file size of the downloaded images spreads over a wide range: 2.78% of them are smaller than 10 KB, 25.59% are in 10 KB∼50 KB, 67.55% are in 50 KB∼500 KB, and 5.21% are larger than 500 KB.

Since the "quality factor" is not consistently defined for different JPEG compression tools, we use a metric, called *average quantization step* (AQS), to evaluate the compression quality. The AQS is computed by averaging 64 quantization steps from the quantization table of luminance channel. The monotonic relationship between IJG QF and AQS, and that between Photoshop QF and AQS are given in Fig. 4. The distribution of compression qualities of the image set is shown in Fig. 5. We can infer from the figure that 5.2% of the

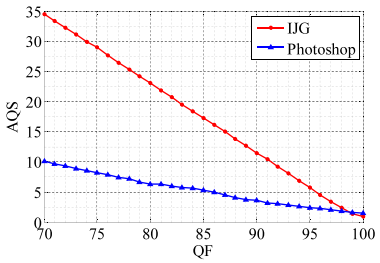


Fig. 4. The relation between the average quantization step (AQS) and the quality factor (QF).

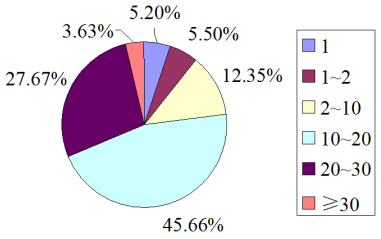


Fig. 5. The distribution of average quantization step (AQS) of the downloaded JPEG images.

images are compressed by a quantization table where all steps are ones, and 5.5% of the images are compressed by a quantization table that corresponds to that of IJG QF=99 (or Photoshop QF from 98 to 100). It indicates that in the age of high-speed Internet, high-quality compressed JPEG images are very common, and they may become much more as Internet bandwidth is getting cheaper.

We also use the 10,000 images of size 512×512 pixels without resizing from BOWS2 image database [40] to play the role of uncompressed images. The reason why we do not use the images in uncompressed format from Internet search engines is that the search engines do not provide the ground-truth information of the image compression history.

3) Evaluation Metrics and Results: We define the *TP* (true positive) and the *TN* (true negative) respectively as the ratio of the decompressed images being correctly classified, and the uncompressed images being correctly classified. From (14), we know that *TP* and *TN* depend on the classification threshold R. On one hand, we report the performance when $R = R_{best}$, where R_{best} is selected to maximize the *ACC* (accuracy), which can be simply computed as the amount of correctly identified decompressed images and uncompressed images over the total amount of test images. Note that in this case, R_{best} may depend on the image dataset. On the other hand, we report the performance when $R = 0.5$, which may be a blind but reasonable criterion due to the majority rule.

The performances with two different criteria on selecting R are reported in Table X, where we also include the value of R_{best} for each method. In accordance with the performance reported in the previous section, our method is the most accurate one under both criteria. It is not surprising that the performance under $R = 0.5$ is inferior to that under $R_{best} \neq 0.5$. The performance drop is not obvious in our method, which indicates that in practical scenario where

TABLE X
PERFORMANCE ON INTERNET IMAGE CLASSIFICATION

		128 × 128			64 × 64			32 × 32		
		Lai's	Luo's	Ours	Lai's	Luo's	Ours	Lai's	Luo's	Ours
R_{best}		0.25	0.63	0.63	0.11	0.50	0.47	0.09	0.50	0.34
$R = R_{best}$	*ACC*	91.71	94.68	**98.98**	95.24	94.18	**99.11**	97.81	94.08	**98.98**
	TP	87.19	92.43	99.03	93.59	92.79	99.01	96.57	91.74	98.84
	TN	98.49	98.06	98.92	97.71	96.26	99.26	99.67	97.60	99.19
$R = 0.5$	*ACC*	87.57	94.56	**98.88**	88.14	94.18	**99.07**	89.52	94.08	**98.40**
	TP	79.30	93.53	99.53	80.23	92.79	98.87	82.53	91.74	97.49
	TN	99.98	96.10	97.89	100.00	96.26	99.38	100.00	97.60	99.78

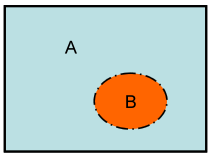


Fig. 6. Illustration of feasible forgery detection scenarios.

training is not available, $R = 0.5$ may be a reasonable threshold.

B. Forgery Detection

The second application of our method is image tampering detection. Once an image has inconsistency in JPEG compression history among different parts, possible forgery may be detected. Suppose an image forgery is composed of two parts as illustrated in Fig. 6. Part A is from a decompressed JPEG image, while Part B is inserted from another image. Even if Part A is decompressed from a high-quality compressed JPEG image, our method is capable of detecting image forgery that belongs to one of the following cases.

Forgery Case A: Part B is from an uncompressed image.

Forgery Case B: Part B is synthesized through a computer graphics rendering or uncompressed image-based synthesis technique.

1) Forgery Detection Algorithm: Given a color test image, we first extract its luminance channel, and then perform JPEG identification independently on non-overlapping $B \times B$-pixel macro-blocks of the luminance channel. Considering a good trade-off between detection sensitivity and accuracy, we use $B = 32$ for forgery detection.

Through the macro-block based detection, each macro-block will be identified as uncompressed or JPEG decompressed. Ideally, we would expect the detection outcomes from a clean image to be consistent over the entire image. In contrast, some distinctive patterns should appear in the manipulated regions of a tampered image.

We use two examples to demonstrate the detection results. When a JPEG decompressed part has been identified, we show its original brightness. Otherwise, we use a dark macro-block to replace the identified uncompressed part.

We give an example of Forgery Case A. Fig. 7 (a) is a decompressed JPEG image, whose previous JPEG quality factor is IJG QF=90. A license plate image, in uncompressed format, as shown in Fig. 7(e), is resized and inserted into the decompressed image. The resulting image is saved in uncompressed format and shown in Fig. 7(i). The detection results provided by Lai's method, Luo's method, and our method for these three kinds of images are respectively shown in Fig. 7(b) to (d), (f) to (h), and (j) to (l). It can be observed that both Luo's method and our method can differentiate the decompressed image from the uncompressed image, and they can recognize the tampered part well. Lai's method can detect the tampered part; however, it makes some false positives on the non-tampered part, especially those macro-blocks in smooth regions.

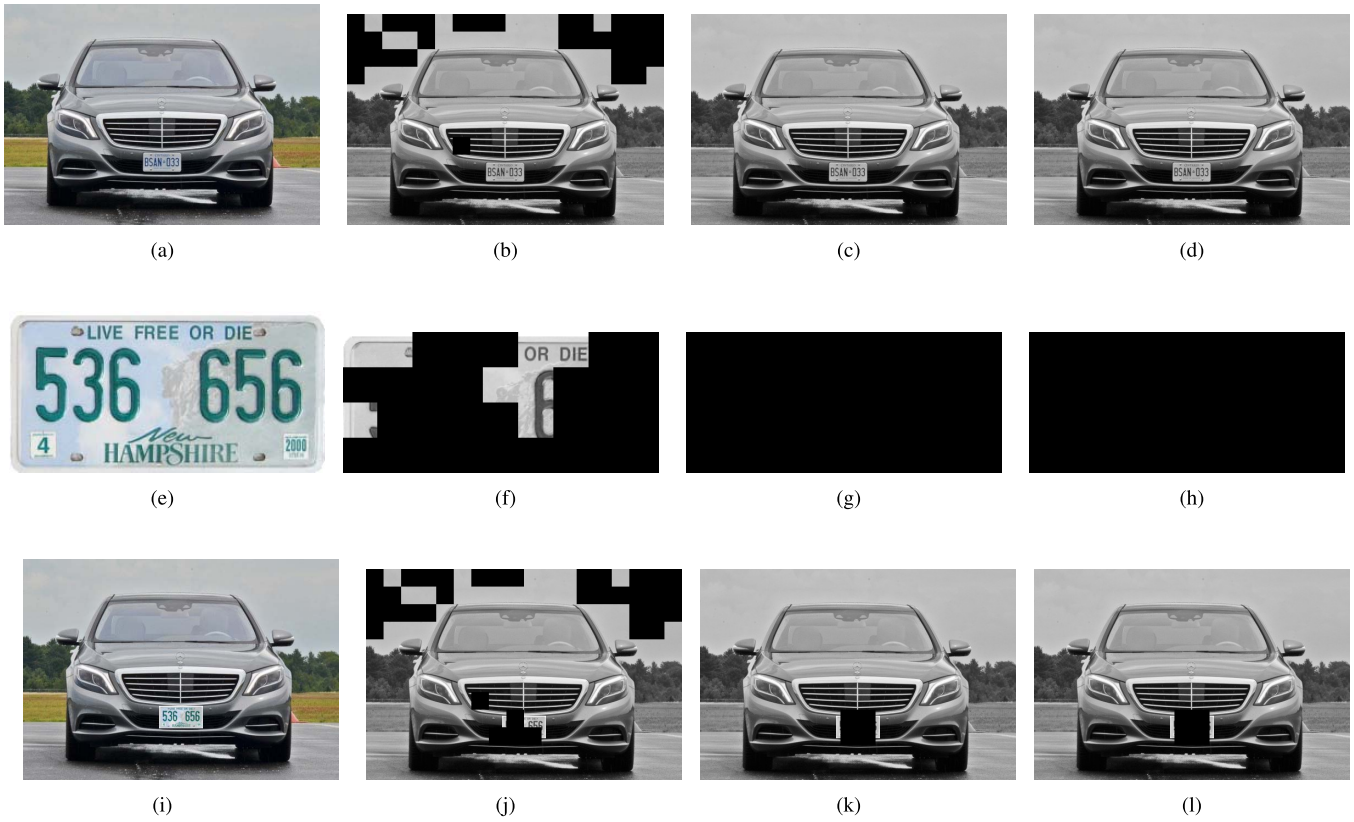
An example of Forgery Case B is given in Fig. 8, where Fig. 8(a) is a decompressed JPEG image, whose previous JPEG quality factor is Photoshop QF=95. An apple in the image is removed by the "content-aware-filling" function of Photoshop CS5. The resulting image is saved in uncompressed format and shown in Fig. 8(d). The detection results to these images provided by Lai' method, Luo's method, and our method are respectively shown in Fig. 8(b) to (d), and (f) to (h). It can be observed that both Lai's method and our method can detect the tampered part; however, Lai's method makes some false positives on the non-tampered part. Luo's method wrongly labels most of the parts as uncompressed and performs the worst among the three methods, which is in accordance with the performance reported in Table V for Photoshop QF=95.

2) Test Image Set: Currently there is no off-the-shelf forgery dataset on decompressed JPEG images. Therefore we create two sets based on the 10,000 images of size 512×512 pixels from BOWS2 database [40] for our experiments, and they are available to the research community.[1]

3) Set A: To simulate the scenario of Forgery Case A, we first compress the image with random IJG QFs, ranging from 75 to 100, then decompress and save it in uncompressed format. The decompressed images are served as negative samples. Next, inspired by the forgery creation process by Fontani *et al.* [39], we cut a portion (from position $(193, 193)$ to $(256, 256)$ with 64×64 pixels) of each uncompressed image and paste it into its decompressed counterpart, exactly in the same position. No perceptual clue can be found in the composite image.

4) Set B: To simulate the scenario of Forgery Case B, we first compress the image with Photoshop and then we decompress and save it in uncompressed format.

[1] available at http://ist.sysu.edu.cn/BOWS2F/idb.html.

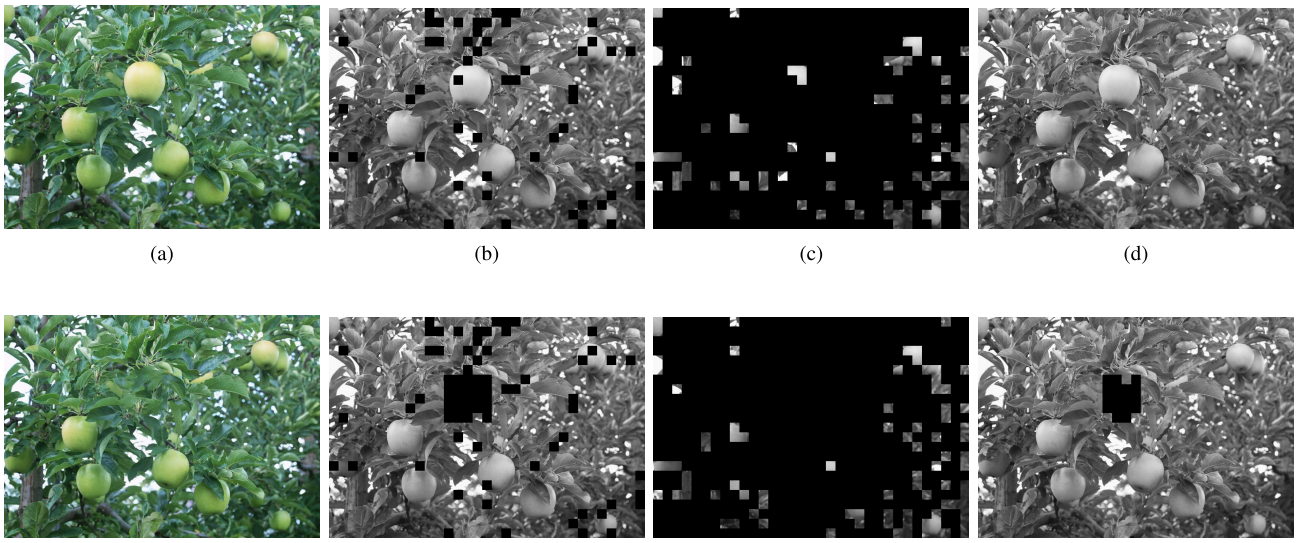


(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Fig. 7. Detection of image-splicing forgery. (a) Decompressed JPEG image. (b) Detection by Lai's method on the decompressed JPEG image. (c) Detection by Luo's method on the decompressed JPEG image. (d) Detection by our method on the decompressed JPEG image. (e) Uncompressed image. (f) Detection by Lai's method on the uncompressed image. (g) Detection by Luo's method on the uncompressed image. (h) Detection by our method on the uncompressed image. (i) Spliced image. (j) Detection by Lai's method on the spliced image. (k) Detection by Luo's method on the spliced image. (l) Detection by our method on the spliced image.



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

(e) (f) (g) (h)

Fig. 8. Detection of content-aware-filling forgery. (a) Decompressed JPEG image. (b) Detection by Lai's method on the decompressed JPEG image. (c) Detection by Luo's method on the decompressed JPEG image. (d) Detection by our method on the decompressed JPEG image. (e) Content-aware-filling image. (f) Detection by Lai's method on the content-aware-filling image. (g) Detection by Luo's method on the content-aware-filling image. (h) Detection by our method on the content-aware-filling image.

Next, we perform a "content-aware-filling" operation in a 100×100-pixels region (from position $(181, 181)$ to $(280, 280)$) of the image. We use Photoshop QF=95 and QF=90 to simulate high-quality compression. A sample image and its tampered counterpart are demonstrated in Fig. 9.

Fig. 9. An example of a pristine image (left) and its tampered counterpart (right).

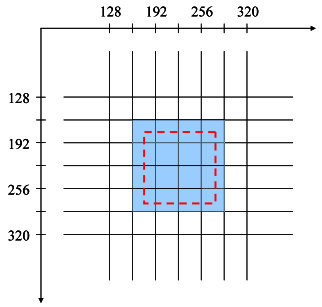


Fig. 10. Illustration of the tampered region (the block with dash line) for each image in Set B.

5) Evaluation Metrics and Results: In Set A, since the composite image is of size 512×512, there will be an amount of 256 macro-blocks of size 32×32. Among them, exactly 4 macro-blocks are from the uncompressed image. When all 252 macro-blocks in the outer region of the composite image are identified as decompressed, and at least 2 out of the 4 macro-blocks in the inner tampered region are identified as uncompressed, we regard the image as being correctly identified. The decision criterion to the tampered part is based on the majority rule as being used in Section V-A.

In set B, the tampered 100×100 region is not aligned with the 32×32 macro-block grid, as illustrated in Fig. 10. A number of 16 macro-blocks may be involved in the manipulation, but only 4 of them are fully covered by the tampered region. As a result, we regard the tampered image as being correctly detected when all 240 macro-blocks in the outer region are identified as decompressed, and at least 2 out of the fully-covered 4 macro-blocks are identified as uncompressed.

For non-tampered image, when all 256 macro-blocks are identified as decompressed, we regard the image as being correctly detected.

The detection results are shown in Table XI, where the detection accuracy of the tampered images and the non-tampered images in Set A and Set B are respectively reported. Our method performs the best. It seems that our method is more effective on Set B; however, we cannot interpret that the content-aware-filling is easier to be detected. In fact, our method has an advantage on the images which are compressed with a higher compression quality in Set B. Besides, the tampering is performed in a larger region in Set B. Luo's method performs better on Set A and the non-tampered images in Set B with a lower Photoshop QF.

TABLE XI
DETECTION RESULTS (IN %) ON FORGERY DETECTION

	Image Type	Lai's	Luo's	Ours
Set A	Non-tampered	17.27	75.30	**84.58**
	Tampered	16.98	71.55	**84.40**
Set B	Non-tampered (Photoshop QF=95)	51.88	15.06	**99.98**
	Non-tampered (Photoshop QF=90)	68.26	87.62	**99.95**
	Tampered (Photoshop QF=95)	0.15	0.00	**95.37**
	Tampered (Photoshop QF=90)	2.07	1.46	**91.59**

Lai's method performs poor on Set A and the tampered images in Set B. These results conform to the two examples shown in Fig. 7 and Fig. 8.

VI. CONCLUSION

In this paper, we propose a method to reveal the traces of JPEG compression. The proposed method is based on analyzing the forward quantization noise, which is obtained by quantizing the block-DCT coefficients with a step of one. A decompressed JPEG image has a lower noise variance than its uncompressed counterpart. Such an observation can be derived analytically. The main contribution of this work is to address the challenges posed by high-quality compression in JPEG compression identification. Specifically, our method is able to detect the images previously compressed with IJG QF=99 or 100, and Photoshop QF from 90 to 100. Experiments show that high-quality compressed images are common on the Internet, and our method is effective to identify them. Besides, our method is robust to small image size and color sub-sampling in chrominance channels. The proposed method can be applied to Internet image classification and forgery detection with relatively accurate results. It should be noted that the proposed method is limited to discriminating uncompressed images from decompressed ones which have not undergone post-processing. Our future studies will be on trying to extend the noise analysis to other forensics tasks, *i.e.*, identifying the resized decompressed JPEG images such as the images presented in IEEE IFS (Information Forensics and Security) Image Forensic Challenge [41].

APPENDIX A
PROOF OF PROPOSITION 1

Assume quantizing a random variable S with a step q. The resulting quantization noise is denoted by ϵ. Denote the characteristic function (CF) of S as $\Psi_S(t)$. It has been shown in [42] that the probability density function (PDF) of the quantization noise can be expressed in a series form as

$$f_\epsilon(x) = \frac{1}{q} + \frac{1}{q} \sum_{k \in \mathbb{Z}, k \neq 0} \Psi_S(\frac{2\pi k}{q}) e^{-j\frac{2\pi k}{q}x}, \quad x \in [-\frac{q}{2}, \frac{q}{2}). \tag{15}$$

When S follows a zero-mean Laplacian distribution with parameter λ, *i.e.*, $S \sim \mathcal{L}(0, \lambda)$, its CF is $\Psi_S(t) = \frac{\lambda^2}{\lambda^2 + t^2}$, $t \in (-\infty, \infty)$. As a result, we can express the PDF of the

quantized-Laplacian distribution as

$$f_\epsilon(x)=\frac{1}{q}+\frac{2}{q}\sum_{k=1}^{\infty}\frac{q^2\lambda^2}{q^2\lambda^2+4\pi^2k^2}\cos(\frac{2\pi k}{q}x),\quad x\in[-\frac{q}{2},\frac{q}{2}). \tag{16}$$

The variance can be obtained by

$$\begin{aligned}\sigma_\epsilon^2 &= \int_{-\frac{q}{2}}^{\frac{q}{2}} x^2 f_\epsilon(x)dx \\ &= \frac{q^2}{12}+\frac{q^2}{\pi^2}\sum_{k=1}^{\infty}\frac{(-1)^k}{k^2}\frac{q^2\lambda^2}{q^2\lambda^2+4\pi^2k^2}.\end{aligned} \tag{17}$$

Note that the first term of the second equation in (17) is $\frac{q^2}{12}$, which equals to the variance of a uniform distribution with support on $[-\frac{q}{2},\frac{q}{2})$. The second term is a convergent alternating series with decreasing absolute value with respect to k. As a result, $\sigma_\epsilon^2\le\frac{q^2}{12}$, and this completes the proof.

Appendix B
Derivation of Proposition 2

When $\upsilon\sim\mathcal{N}(0,\sigma^2)$, its CF is $\Psi_Z(t)=e^{-\frac{1}{2}\sigma^2t}$, $t\in(-\infty,\infty)$. According to (15), when $q=1$, the PDF of the quantized Gaussian is

$$f_{n_\upsilon}(x)=1+2\sum_{k=1}^{\infty}e^{-2\pi^2k^2\sigma^2}\cos(2\pi kx),\quad x\in[-\frac{1}{2},\frac{1}{2}). \tag{18}$$

Its variance is given by

$$\sigma_{n_\upsilon}^2=\frac{1}{12}+\frac{1}{\pi^2}\sum_{k=1}^{\infty}\frac{(-1)^k}{k^2}e^{-2\pi^2k^2\sigma^2}. \tag{19}$$

Note that the second term of (19) is a convergent alternating series with decreasing absolute value with respect to k. As a result, $\sigma_{n_\upsilon}^2$ is upper bounded by $\sigma_{n_\upsilon}^2\le C_0=\frac{1}{12}=0.0833$ for any value of σ^2. When we respectively assign $\sigma^2=C_0=0.0833$ and $\sigma^2=C_1=0.0638$, and use a sufficient large number of k in (19), we can numerically obtain $\sigma_{n_\upsilon}^2=C_1=0.0638$ and $\sigma_{n_\upsilon}^2=C_2=0.0548$.

Appendix C
Proof of Proposition 3

When $q_u^{(1)}=1,\forall u$, according to Property 1 and Proposition 1, we can obtain $\sigma^2_{y_u^{(1)}}\le C_0=\frac{1}{12}=0.0833,\forall u$. The auxiliary noise $x^{(1)}$ is the inverse DCT transform of the quantization noise $y^{(1)}$ [31]. According to the central limit theorem, $x^{(1)}\sim\mathcal{N}(0,\sigma^2_{x^{(1)}})$. Its variance is bounded by

$$\sigma^2_{x^{(1)}}\le\max_u\{\sigma^2_{y_u^{(1)}}\}\le C_0=0.0833. \tag{20}$$

The rounding noise $x^{(1\to2)}$ is arising from negatively rounding off the auxiliary noise $x^{(1)}$ [31]. Hence we know $x^{(1\to2)}\sim\mathcal{Q}^{\mathcal{N}}(\sigma^2_{x^{(1)}},1)$. Based on the second condition of Proposition 2, the variance of $x^{(1\to2)}$ is bounded by

$$\sigma^2_{x^{(1\to2)}}\le C_1=0.0638. \tag{21}$$

The auxiliary noise $y^{(1\to2)}$ is the DCT transform of the rounding noise $x^{(1\to2)}$ [31]. According to the central limit theorem, $y^{(1\to2)}\sim\mathcal{N}(0,\sigma^2_{y^{(1\to2)}})$. Its variance is bounded by

$$\sigma^2_{y^{(1\to2)}}\le\max_m\{\sigma^2_{x_m^{(1\to2)}}\}\le C_1=0.0638. \tag{22}$$

This completes the proof of the first case of Proposition 3.

When the condition $q_u^{(1)}=1,\forall u$ is not satisfied, (22) does not hold. However, based on Proposition 2, for any value of σ^2, we know that the variance of $x^{(1\to2)}$ is bounded by

$$\sigma^2_{x^{(1\to2)}}\le C_0=0.0833. \tag{23}$$

Similar to (22), we can arrive

$$\sigma^2_{y^{(1\to2)}}\le\max_m\{\sigma^2_{x_m^{(1\to2)}}\}\le C_0=0.0833. \tag{24}$$

This completes the proof of the second case of Proposition 3.

Acknowledgment

The authors appreciate Dr. Teddy Furon and Dr. Patrick Bas for their permission on making the forgery image database available for the public. The authors also appreciate Guangdong Key Lab of Information Security Technology for hosting the forgery image database.

References

[1] A. Piva, "An overview on image forensics," *ISRN Signal Process.*, vol. 2013, pp. 1–22, Nov. 2013.
[2] M. C. Stamm, M. Wu, and K. J. R. Liu, "Information forensics: An overview of the first decade," *IEEE Access*, vol. 1, pp. 167–200, May 2013.
[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] H. Farid, "Exposing digital forgeries from JPEG ghosts," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 1, pp. 154–160, Mar. 2009.
[6] T. Pevný 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.
[7] Y.-L. Chen and C.-T. Hsu, "Detecting doubly compressed images based on quantization noise model and image restoration," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, Oct. 2009, pp. 1–6.
[8] F. Huang, J. Huang, and Y. Q. Shi, "Detecting double JPEG compression with the same quantization matrix," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 848–856, Dec. 2010.
[9] T. Gloe, "Demystifying histograms of multi-quantised DCT coefficients," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2011, pp. 1–6.
[10] S. Lai and R. Böhme, "Block convergence in repeated transform coding: JPEG-100 forensics, carbon dating, and tamper detection," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2013, pp. 3028–3032.
[11] W. Luo, Z. Qu, J. Huang, and G. Qiu, "A novel method for detecting cropped and recompressed image block," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 2. Apr. 2007, pp. II-217–II-220.
[12] Z. Qu, W. Luo, and J. Huang, "A convolutive mixing model for shifted double JPEG compression with application to passive image authentication," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Las Vegas, NV, USA, Mar./Apr. 2008, pp. 1661–1664.
[13] Y.-L. Chen and C.-T. Hsu, "Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 2, pp. 396–406, Jun. 2011.
[14] Q. Liu, "Detection of misaligned cropping and recompression with the same quantization matrix and relevant forgery," in *Proc. 3rd ACM Int. Workshop Multimedia Forensics Intell.*, 2011, pp. 25–30.

[15] T. Bianchi, A. Piva, and F. Perez-Gonzalez, "Near optimal detection of quantized signals and application to JPEG forensics," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, Guangzhou, China, Nov. 2013, pp. 168–173.

[16] D. Fu, Y. Q. Shi, and W. Su, "A generalized Benford's law for JPEG coefficients and its applications in image forensics," *Proc. SPIE, Secur., Steganogr., Watermarking Multimedia Contents IX*, vol. 6505, pp. 65051L-1–65051L-11, Feb. 2007.

[17] J. Fridrich, M. Goljan, and R. Du, "Steganalysis based on JPEG compatibility," *Proc. SPIE, Multimedia Syst. Appl. IV*, vol. 4518, pp. 275–280, Nov. 2001.

[18] R. N. Neelamani, R. de Queiroz, Z. Fan, S. Dash, and R. G. Baraniuk, "JPEG compression history estimation for color images," *IEEE Trans. Image Process.*, vol. 15, no. 6, pp. 1365–1378, Jun. 2006.

[19] S. Ye, Q. Sun, and E.-C. Chang, "Detecting digital image forgeries by measuring inconsistencies of blocking artifact," in *Proc. IEEE Int. Conf. Multimedia Expo*, Beijing, China, Jul. 2007, pp. 12–15.

[20] T. C.-I. Lin, M.-K. Chang, and Y.-L. Chen, "A passive-blind forgery detection scheme based on content-adaptive quantization table estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 4, pp. 421–434, Apr. 2011.

[21] F. Galvan, G. Puglisi, A. R. Bruna, and S. Battiato, "First quantization matrix estimation from double compressed JPEG images," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 8, pp. 1299–1310, Aug. 2014.

[22] M. C. Stamm and K. J. R. Liu, "Anti-forensics of digital image compression," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 1050–1065, Sep. 2011.

[23] W. Fan, K. Wang, F. Cayre, and Z. Xiong, "JPEG anti-forensics with improved tradeoff between forensic undetectability and image quality," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 8, pp. 1211–1226, Aug. 2014.

[24] 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.

[25] S. Lai and R. Böhme, "Countering counter-forensics: The case of JPEG compression," in *Proc. 13th Int. Conf. Inf. Hiding Workshop* (Lecture Notes in Computer Science), vol. 6958, Prague, Czech Republic, 2011, pp. 285–298.

[26] H. Li, W. Luo, and J. Huang, "Countering anti-JPEG compression forensics," in *Proc. 19th IEEE Int. Conf. Image Process.*, Sep./Oct. 2012, pp. 241–244.

[27] 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.

[28] S. Minami and A. Zakhor, "An optimization approach for removing blocking effects in transform coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 2, pp. 74–82, Apr. 1995.

[29] K. T. Tan and M. Ghanbari, "Blockiness detection for MPEG2-coded video," *IEEE Signal Process. Lett.*, vol. 7, no. 8, pp. 213–215, Aug. 2000.

[30] J. Fridrich, "Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes," in *Proc. 6th Int. Workshop Inf. Hiding*, vol. LNCS 3200. 2005, pp. 67–81.

[31] B. Li, T.-T. Ng, X. Li, S. Tan, and J. Huang, "JPEG noises beyond the first compression cycle," Dept. College Inf. Eng., Shenzhen Univ., Shenzhen, China, Tech. Rep. TR2014-001, 2014. [Online]. Available: http://arxiv.org/abs/1405.7571

[32] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, no. 4, pp. 30–44, 1991.

[33] A. C. Popescu and H. Farid, "Statistical tools for digital forensics," in *Information Hiding*. Berlin, Germany: Springer-Verlag, 2005, pp. 128–147.

[34] T. Pevný, 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.

[35] P. Bas, T. Filler, and T. Pevný, "'Break our steganographic system': The ins and outs of organizing BOSS," in *Proc. 13th Int. Conf. Inf. Hiding Workshop* (Lecture Notes in Computer Science), vol. 6958. Prague, Czech Republic, 2011, pp. 59–70.

[36] *NRCS Photo Gallery*. [Online]. Available: http://photogallery.nrcs.usda.gov/res/sites/PhotoGallery/index.html, accessed Jan. 2015.

[37] G. Schaefer and M. Stich, "UCID: An uncompressed color image database," *Proc. SPIE, Storage Retr. Methods Appl. Multimedia*, vol. 5307, pp. 472–480, Dec. 2003.

[38] *Independent JPEG Group Library (IJG)*. [Online]. Available: http://www.ijg.org, accessed Jan. 2015.

[39] M. Fontani, T. Bianchi, A. De Rosa, A. Piva, and M. Barni, "A framework for decision fusion in image forensics based on Dempster–Shafer theory of evidence," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 4, pp. 593–607, Apr. 2012.

[40] T. Furon and P. Bas, "Broken arrows," *EURASIP J. Inf. Secur.*, vol. 2008, p. 597040, Sep. 2008.

[41] Information Forensics and Security Technical Committee. (2013). *IEEE IFS Image Forensic Challenge*. [Online]. Available: http://ifc.recod.ic.unicamp.br/fc.website/index.py

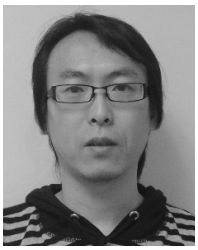
[42] A. B. Sripad and D. Snyder, "A necessary and sufficient condition for quantization errors to be uniform and white," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 25, no. 5, pp. 442–448, Oct. 1977.

Bin Li (S'07–M'09) received the B.E. degree in communication engineering and the Ph.D. degree in communication and information systems from Sun Yet-sen University, Guangzhou, China, in 2004 and 2009, respectively.

He is currently an Associate Professor with Shenzhen University, Shenzhen, China, where he joined in 2009. He is also a member of the Shenzhen Key Laboratory of Advanced Communications and Information Processing. From 2007 to 2008, he was a Visiting Scholar with the New Jersey Institute of Technology, Newark, NJ, USA. His current research interests include image processing, multimedia forensics, and pattern recognition.

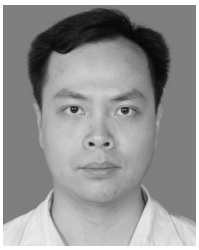
Tian-Tsong Ng received the B.Eng. (Hons.) degree in electrical engineering from the University of Malaya, Kuala Lumpur, Malaysia, in 1998, the M.Phil. degree in signal processing from Cambridge University, Cambridge, U.K., in 2001, and the Ph.D. degree in electrical engineering from Columbia University, New York, NY, USA, in 2007.

He is currently a Research Scientist with the Institute for Infocomm Research, Singapore. His research interest lies in the application of advanced signal processing methods to uncover the structure of image formation for solving problems in computer vision, computer graphics, and image forensics.



Xiaolong Li received the B.S. degree from Peking University, Beijing, China, in 1999, the M.S. degree from École Polytechnique, Palaiseau, France, in 2002, and the Ph.D. degree in mathematics from the École Normale Supérieure de Cachan, Cachan, France, in 2006.

He is currently a Researcher with the Institute of Computer Science and Technology, Peking University, where he was a Post-Doctoral Fellow from 2007 to 2009. His research interests are image processing and information hiding.



Shunquan Tan (M'10) received the B.S. degree in computational mathematics and applied software and the Ph.D. degree in computer software and theory from Sun Yat-sen University, Guangzhou, China, in 2002 and 2007, respectively.

He is currently a Lecturer with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. He is also a member of the Shenzhen Key Laboratory of Media Security. His current research interests include steganography, steganalysis, multimedia forensics, and deep machine.

Jiwu Huang (M'98–SM'00) received the B.S. degree from Xidian University, Xi'an, China, in 1982, the M.S. degree from Tsinghua University, Beijing, China, in 1987, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 1998.

He is currently a Professor with the College of Information Engineering, Shenzhen University, Shenzhen, China. Before joining Shenzhen University, he was with the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China. His current research interests include multimedia forensics and security. He is also a member of the IEEE Circuits and Systems Society Multimedia Systems and Applications Technical Committee and the IEEE Signal Processing Society Information Forensics and Security Technical Committee. He served as an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY and a General Cochair of the IEEE International Workshop on Information Forensics and Security 2013.