JPEG Error Analysis and Its Applications to Digital Image Forensics

Weiqi Luo, *Member, IEEE,* Jiwu Huang, *Senior Member, IEEE,* and Guoping Qiu, *Member, IEEE*

Abstract

JPEG is one of the most extensively used image formats. Understanding the inherent characteristics of JPEG may play a useful role in digital image forensics. In this paper, we introduce JPEG error analysis to the study of image forensics. The main errors of JPEG include quantization, rounding and truncation errors. Through theoretically analyzing the effects of these errors on single and double JPEG compression, we have developed three novel schemes for image forensics including identifying whether a bitmap image has previously been JPEG compressed, estimating the quantization steps of a JPEG image and detecting the quantization table of a JPEG image. Extensive experimental results show that our new methods significantly outperform existing techniques especially for the images of small sizes. We also show that the new method can reliably detect JPEG image blocks which are as small as 8×8 pixels and compressed with quality factors as high as 98. This performance is important for analyzing and locating small tampered regions within a composite image.

Index Terms

Digital Forensics, JPEG History, Error Analysis, Double JPEG Compression

This work is supported by the NSFC (60633030), 973 Program (2006CB303104) and China Postdoctoral Science Foundation (20080440795).

W. Luo and J. Huang are with the school of information science and technology of Sun Yat-Sen University and Guangdong Key Laboratory of Information Security Technology, Guangzhou, China, 510275.

G. Qiu is with the School of Computer Science University of Nottingham Jubilee Campus, Nottingham NG8 1BB, United Kingdom.

E-mail: isshjw@mail.sysu.edu.cn

I. INTRODUCTION

With the rapid development of image processing technology, it is getting easier to tamper with digital images without leaving any obvious visual trace. Nowadays, seeing is no longer believing [1]. Image forgery, like any other illegal and pernicious activities, could cause serious harms to society. Therefore, verifying the authenticity of digital images has become a very important issue.

Digital watermarks and signatures have been proposed as potential solutions to digital image authentication [2], [3]. However, both of the methods need some proactive processing such as inserting an imperceptible watermark or attaching a signature at the time of the data's creation in order to facilitate tampering detection at a later time. Thus they are regarded as active methods. Recently, a passive technique has evolved quickly [4], [5]. The technique assumes that the original multimedia has some inherent features introduced by the various imaging devices and/or processing. By analyzing how the multimedia data is acquired and processed, we can answer some forensic questions, such as where is the data coming from, whether it is an original one or not, and what tampering operations had been done previously. Compared with prior active methods, the new method does not rely on any extra information such as a watermark or a signature. Therefore, it is passive and completely blind.

It is well-known that JPEG is a commonly used compression standard and has been found in many applications. For instance, most digital cameras on the market export this file format, and most image editing software such as Adobe Photoshop, GIMP, support the operation of JPEG compression. Therefore, analyzing this type of images may play a useful role in image forensics. However, JPEG images are sometimes processed or stored as bitmaps. In that case, we will have no knowledge about whether the bitmap comes from a JPEG image, a raw bitmap image or any other formats. Identifying whether a bitmap has previously been JPEG compressed and if so what quantization table has been used is a crucial first step for many image forensic algorithms. For instance, the algorithms for double JPEG compression detection [6], [7], "shift-recompression" detection [8], tampered regions localization in composite images [9], [10], steganalysis of YASS (Yet Another Steganographic Scheme) [11], [12] *et al.* usually have to first determine whether a given image is a JPEG image or not before detection, and some of the algorithms such as [8], [12] are dependent on the quantization steps/table used in the previous JPEG compression. Moreover, since different cameras typically employ different quantization tables [13], so do different image editing tools as well (*e.g.* the tables used in Photoshop are quite different from those used in GIMP), quantization steps/table estimation can be also used for image's source identification.

So far, just a few effective methods *e.g.* [14], [15], [16] have been proposed for the three forensic

issues *i.e.* identifying JPEG images, estimating quantization steps and detecting quantization tables from a Bitmap image. In [14], Fan *et al.* proposed a method to determine whether a bitmap image has been JPEG compressed by detecting the blocking boundary strength in the spatial domain and further to extract the quantization table using MLE (maximum likelihood estimation) method. In [15], Fridrich *et al.* presented a method to extract the quantization steps from a bitmap image by measuring the compatibility of DCT coefficients for all the candidate steps for quantization. In [16], Fu *et al.* applied Bendford's law (first digit law) on the *DCT* coefficients to estimate the JPEG quantization table and detect double JPEG compressed images. However, most existing methods assume that the test images are large enough to gather sufficient statistics. In many forensic scenarios, the tampered region within an image may just a small patch, such as the face of a person, some important numbers/characters *et al.*. In such cases, the existing methods would usually fail. Taking JPEG images identification for example, it is impossible to detect an 8×8 image block using all the previous methods. Moreover, some methods, such as Benford's law based works [16], are machine learning based, which means that they need to train a classifier under the well designed situation prior to detection. Please note that there are many other literatures related to JPEG image enhancement *e.g.* [17], [18], [19], [20], [21], [22], [23], [24], [25]. However, these methods usually investigate the JPEG images coded at low bit-rates and aim to reduce the blocking artifacts between the block boundaries and ringing effect around the content edges due to the lossy JPEG compression. Some of them *e.g.* [17], [18], [19], [20], [21], [22] may be extended to identify JPEG images, however, the performances are very poor based on our experiments as shown in Section III-A. We also note that there are several reported methods *e.g.* [6], [7], [26], [27], [28], [29], [30], for other forensics/steganlysis issues which tried to identify the double JPEG compressed images and/or further estimate the primary quantization table. In such cases, the input images of these algorithms are JPEG images rather than bitmap images, which means that we can obtain all the parameters of the last JPEG compression from the file header, and therefore there methods will not be suitable for the proposed forensic situations.

In this paper, we first analyze the main sources of errors introduced by JPEG compression and their relationships to JPEG recompression, and then propose three novel methods for the estimation of JPEG history. The new methods are derived based on the following observations:

- The *AC* coefficients of an image will increase in the range of $(-1, +1)$ while decrease significantly in the union regions of $(-2, -1]$ and $[+1, +2)$ after JPEG compression with quantization steps that are equal to or larger than 2. Based on this observation, a 1-D feature can be obtained to differentiate between uncompressed and JPEG images.

- Due to rounding error, the *AC* coefficients of a JPEG decompressed image for a given frequency component will not present exactly at the multiples of the corresponding quantization step, but will spread around them at an interval between $(-1, +1)$ with a high probability. By removing such rounding effect, an effective feature can be obtained for quantization step estimation.
- When a JPEG image is JPEG compressed again, most original pixels will be better preserved if the recompression uses the same quantization table as that used for the original JPEG. This is true even when the recompression quality factor is set to 100 (all quantization steps are equal to 1). Based on this observation, we obtain a simple but very effective method to extract the quantization table from a JPEG decompressed image.

We present theoretic analysis and experimental results which show that our techniques are very effective and perform better than existing methods, especially for the images of small sizes.

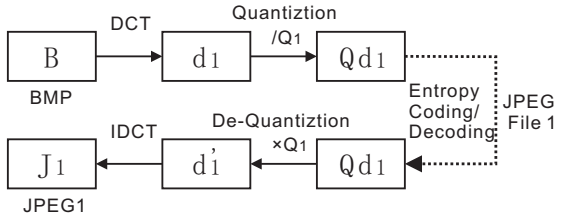
The rest of the paper is organized as follows. Section II theoretically analyzes the errors introduced by JPEG compression and recompression, and then presents three forensic analysis techniques. Section III shows experimental results, and concluding remarks of this paper will be presented in Section IV.

II. ERROR ANALYSIS FOR JPEG COMPRESSION AND RECOMPRESSION

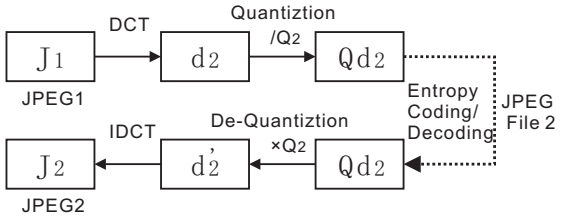
Lossy JPEG compression is a block-based compression scheme. Each non-overlapping 8×8 block within an image is dealt with separately. In the following, we will consider the processes of JPEG compression and recompression on an 8×8 uncompressed block B.

As shown in Fig.1 (a), the image block B is first transformed from the spatial domain into the frequency domain using discrete cosine transform (*DCT*) , and then the frequency coefficients d_1 is quantized by a quantization table Q_1. The error introduced in this stage is called quantization error, the main cause of information loss for the images having not been JPEG compressed before. The quantization coefficients Qd_1 is then further compressed by the lossless entropy encoding. Finally, the resulting bit stream is combined with a header to form a JPEG file.

In JPEG decompression, the compressed file is first entropy decoded to recover the quantized coefficients Qd_1 exactly, which are then multiplied by the table Q_1 to obtain the de-quantized coefficients d_1'. And then the Inverse *DCT* (*IDCT*) is performed on d_1' to get the pixel values in real-valued representation. Since gray-scale images are typically stored with 8 bits per pixel, which can represent 256 different intensities, the resulting real values will be rounded and truncated to the nearest integers in the range of $[0, 255]$. Two errors will be generated in this stage, namely rounding and truncation errors. Please note that the rounding error usually occurs in every 8×8 block, while the truncation error does not. In the



(a) Single JPEG



(b) Double JPEG

Fig. 1. Single and double JPEG compression & decompression

following, we test the truncation error on the following five uncompressed image datasets: Corel (1,096 images coming from the CorelDraw version 10.0 software CD #3), NJIT (3,685 images form New Jersey Institute of Technology, USA), NRCS (1,543 images randomly downloaded from the NRCS photo gallery [31]), SYSU (2,338 images taken by Panasonic DMC-FZ30 and Nikon D300 in TIFF format) and UCID (1,338 images [32]), with a quality factor of 98, 95, 85, 75 and 50, respectively. In all, there are $10,000$ natural color images including (but not limited to) landscapes, people, plants, animals and buildings. We first convert these color images into gray-scale images, and then simulate the JPEG compression and decompression using the Matlab JPEG Toolbox [33] to obtain the de-quantized coefficients d_1' as shown in Fig.1 (a), and then perform the *IDCT* operation on the de-quantized coefficients d_1' to get the real-values representation of the JPEG image without rounding and truncation operations. Finally we compute the percentage of the resulting pixel values that are larger than 255 or smaller than 0 and show the experimental results in Table I.

From the Table I, it is observed that the truncation error are highly dependent on the quality factors used in JPEG compression as well as the tested image datasets. Usually, the smaller the quality factor (larger quantization steps) is, the higher the probability of the truncation error becomes. More importantly, we can see that in natural JPEG images, such error just occurs with a very low probability (less than 1% in most cases as shown in the table). In practice, we can easily eliminate the effect of truncation error by just investigating the unsaturated blocks (those 8×8 blocks without containing pixels with value 0 or

TABLE I

THE AVERAGE PERCENTAGES (%) OF TRUNCATION ERROR IN NATURAL JPEG IMAGES FROM FIVE IMAGE DATASETS WITH QUALITY FACTORS RANGING FROM 50 TO 98.

Quality Factor \ Dataset	Corel	NJIT	NRCS	SYSU	UCID	Avg.
QF=98	0.01	0.33	0.08	0.26	0.33	**0.20**
QF=95	0.04	0.38	0.09	0.27	0.40	**0.24**
QF=85	0.14	0.48	0.11	0.28	0.50	**0.30**
QF=75	0.21	0.45	0.15	0.30	0.56	**0.33**
QF=50	0.30	1.45	0.42	0.95	1.42	**0.91**

255) within an image. Therefore, we focus on the quantization error and rounding error in the following analysis and experiments.

Similarly, the processes of double JPEG compression and decompression are shown in Fig.1 (b).

A. Identifying JPEG Images

As shown in Fig.1, to discriminate the JPEG images (J_1) from uncompressed ones (B), we investigate the histograms of their *DCT* coefficients, namely, the distributions of d_1 and d_2 instead of the spatial features in B and J_1 directly as used in existing method [14]. Please note that the relationships between d_1 and d_2 can be written in the following form.

$$\begin{aligned} d_2 &= DCT(J_1) \\ &= DCT([IDCT(d_1')]) \\ &= DCT(IDCT(d_1') + \Re) \\ &= DCT(IDCT(d_1')) + DCT(\Re) \\ &= d_1' + \varepsilon \\ &= [\tfrac{d_1}{Q_1}] \times Q_1 + \varepsilon \end{aligned} \tag{1}$$

where $[x]$ is the nearest integer function of a real number x. $\Re$ denotes the rounding errors introduced by previous JPEG decompression and $\varepsilon = DCT(\Re)$. Both $\Re$ and ε are 8×8 matrices. Here, we assume that $\Re(i,j)$ is an $i.i.d.$ random variable with uniform distribution in the range of $[-0.5, +0.5)$. According to the Central Limit Theorem, we can conclude that $\varepsilon(i,j)$ is an approximate Gaussian distribution with mean 0 and variance $\frac{1}{12}$, for all $0 \leq i,j \leq 7$.

We consider the *DCT* coefficients for each frequency positions (i,j) independently, where $0 \leq i,j \leq 7$ except the *DC* component $(i,j) = (0,0)$. Let the probability density function (PDF) of $\varepsilon(i,j)$, $d_1(i,j)$,

$d_1'(i,j)$ and $d_2(i,j)$ be $p_\varepsilon(x)$, $p_1(x)$, $p_1'(x)$ and $p_2(x)$, respectively. And let $q = Q_1(i,j)$ be the quantization step. Assume $q \geq 2$, the most common settings used in JPEG compression.

It is well-known that the *AC* coefficients $d_1(i,j)$ in natural images follow an approximate Laplacian distribution [34]. After quantization and de-quantization operations by a step q, however, most values in the histogram of $d_1(i,j)$ will be quantized to the nearest integers kq, where k is an integer which is determined by the image contents *i.e.* the quantized coefficients Qd_1 at a given position (i,j). Therefore, the PDF of the de-quantized coefficients $d_1'(i,j)$ can be formulated as follows.

$$p_1'(x) = \begin{cases} \int_{kq-q/2}^{kq+q/2} p_1(y)\,\mathrm{d}y & x = kq, k \in Z \\ 0 & otherwise \end{cases} \tag{2}$$

The Eq. (2) shows that $p_1'(x)$ is exactly a multiple of the quantization step q. Different from p_1', p_2 is a noise contaminated version of p_1', it will spread around the multiple of q due to the rounding error $\varepsilon(i,j)$. This can be seen though the relationship between d_1' and d_2 as shown in (1).

Note that $E(\varepsilon(i,j)) = 0, \sigma^2(\varepsilon(i,j)) = \frac{1}{12}$, then we have

$$\int_{-1}^{+1} p_\varepsilon(y)\,\mathrm{d}y \geq 99.95\% \tag{3}$$

Combining Eq. (1)$\sim$(3), we obtain the relationship between p_1' and p_2 as follow.

$$\int_{kq-1}^{kq+1} p_2(y)\,\mathrm{d}y \approx p_1'(kq) \quad k \in Z, q \geq 2 \tag{4}$$

Fig.2 illustrates the distributions of $d_1(1,1), d_1'(1,1)$ and $d_2(1,1)$ for the Lena image with a quality factor 85. The quantization step q is equal to 4 in this case. Please note that the distributions for other frequency components have similar results if the quantization step is equal to or larger than 2.

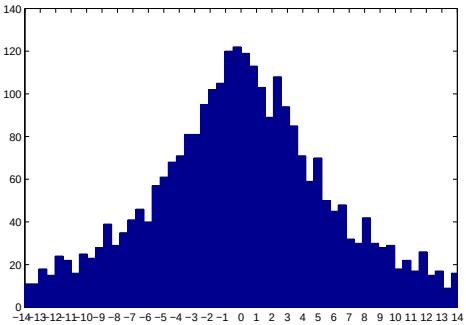
Therefore, the main idea of JPEG image identification is then converted to discriminate the patterns between Fig.2 (b) and (d). To achieve the goal, we consider the percentage of the *AC* coefficients of a given test image in the following two specific regions, that is,

$$R_1 = (-1,+1) \ \ \& \ \ R_2 = (-2,-1] \cup [+1,+2)$$

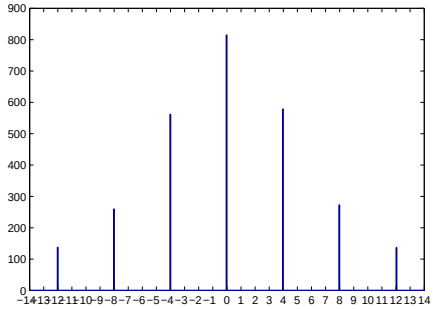
Based on the Eq. (4) and (2), we have

$$\begin{aligned} \int_{R_1} p_2(y)\,\mathrm{d}y &= \int_{-1}^{+1} p_2(y)\,\mathrm{d}y \approx p_1'(0) = \int_{-\frac{q}{2}}^{+\frac{q}{2}} p_1(y)\,\mathrm{d}y \\ &\geq \int_{-1}^{+1} p_1(y)\,\mathrm{d}y = \int_{R_1} p_1(y)\,\mathrm{d}y, \quad q \geq 2 \end{aligned} \tag{5}$$

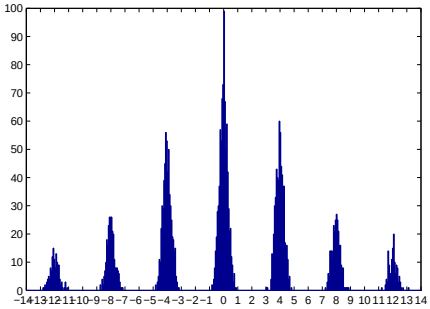
(a) JPEG Lena ($QF = 85$)



(b) Distribution of $d_1(1,1)$



(c) Distribution of $d_1'(1,1)$



(d) Distribution of $d_2(1,1)$

Fig. 2. The distributions of d_1, d_2 and d_1' for Lena image with a quality factor 85 at the position $(1,1)$.

and

$$\begin{aligned} \int_{R_2} p_2(y)\,\mathrm{d}y &= \int_{-2}^{-1} p_2(y)\,\mathrm{d}y + \int_{+1}^{+2} p_2(y)\,\mathrm{d}y \\ &\approx 0 + 0 \le \int_{R_2} p_1(y)\,\mathrm{d}y, \quad q \ge 2 \end{aligned} \tag{6}$$

Therefore, a 1-D feature s can be obtained as

$$s = \frac{\int_{R_2} p_{ac}(y)\ \mathrm{d}y}{\int_{R_1} p_{ac}(y)\,\mathrm{d}y}, \qquad \int_{R_1} p_{ac}(y)\,\mathrm{d}y \neq 0 \tag{7}$$

where p_{ac} denotes the PDF of all the *AC* coefficients of the test image.

Based on Eq. (5) and (6), it is expected that the value $\int_{R_1} p_{ac}(y)\ \mathrm{d}y$ will increase, while the value $\int_{R_2} p_{ac}(y)\ \mathrm{d}y$ will decrease after being quantized by those quantization steps that are equal to or larger than 2, which means that the proposed feature s of a JPEG image will be close to zero and it would be much smaller than that of the corresponding uncompressed one.

B. Estimating Quantization Steps

If without the rounding errors $\Re$, we can recover the de-quantized coefficients $d_1'(i,j)$ from the JPEG decompressed image J_1 exactly based on Eq. (1).

$$d_2 = DCT(J_1) = DCT(IDCT(d_1') + 0) = d_1' \tag{8}$$

The estimation of quantization steps becomes very easy in this case since the values $d_1'(i,j)$ will just present at multiples of the corresponding quantization step $q = Q_1(i,j)$ as illustrated in Fig.2 (c). Therefore, how to remove the effect of the rounding error $\Re$ and recover the de-quantized coefficients d_1' from the noise contaminated version d_2 (Fig.2 (d)) is the key issue in our algorithm.

Based on Eq. (1), we have

$$d_2 = d_1' + DCT(\Re) = d_1' + \varepsilon \tag{9}$$

which shows that the rounding error $\Re$ will lead to an approximate Gaussian noise ε with mean 0 and variance $\frac{1}{12}$ on each *DCT* frequency component of the bitmap image J_1. As illustrated in Fig.2 (d), those peaks in the distribution of $d_2(i,j)$ will spread around the multiples of the quantization step q at a limited interval $(-1,+1)$ according to Eq. (3), and most of them are separated if $q \geq 2$.

When performing the rounding operation on the coefficients d_2, most of the noise contaminated values d_2 will become the original ones (*i.e.* d_1') since

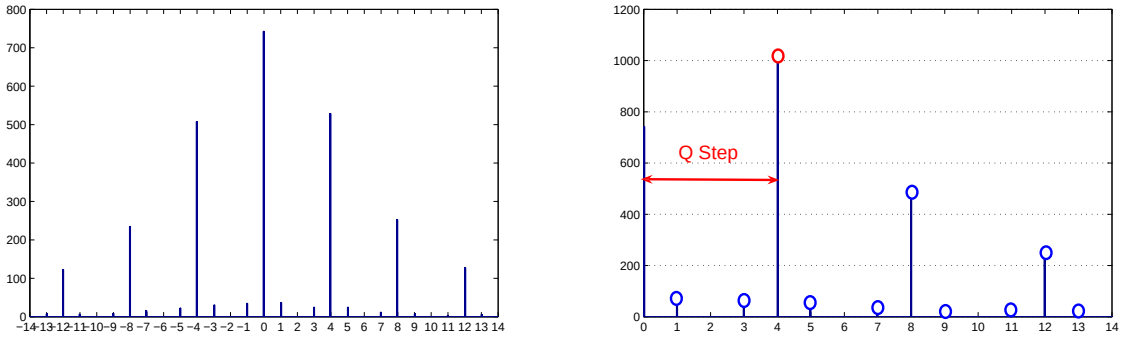
$$[d_2(i,j)] = d_1'(i,j) + [\varepsilon(i,j)] \approx d_1'(i,j) \tag{10}$$

And the probability of the event $[d_2(i,j)] = d_1'(i,j)$ can be calculated by

$$\begin{aligned} P([d_2(i,j)] = d_1'(i,j)) &= P([\varepsilon(i,j)] = 0) \\ &= \int_{-0.5}^{+0.5} p_\varepsilon(y)\,\mathrm{d}y \geq 91.50\% \end{aligned} \tag{11}$$

Fig.3 (a) illustrates the denoised version of Fig.2 (d) after rounding operation. Please refer to Fig.3 (a) and Fig.2 (c), there are still around $8.5\%(1-91.5\%)$ of the coefficients d_2 which will be rounded to the neighborhoods of the multiples of the quantization step (see Eq. (11)). In order to remove such effect of the "ghost" values in the distribution of $[d_2]$ and to extract the quantization step, we first obtain the absolute values of $[d_2]$ and then calculate the histogram H of the resulting values $|[d_2]|$ for each frequency component (i,j) independently, as shown in Fig.3 (b). Then the location of the histogram bin containing the most non-zeros coefficients indicates the quantization step $\hat{q_{i,j}}$.

We know that most DCT coefficients of natural images will be quantized to 0 after JPEG compression. However, around 8.5% of those coefficients will be rounded to its neighborhoods *i.e.* ± 1 according to Eq.



(a) Distribution of $[d_2(1,1)]$ (b) Distribution of $|[d_2(1,1)]|$

Fig. 3. The distributions of $[d_2]$ and $|[d_2]|$ for Lena image with a quality factor 85 at the position $(1,1)$.

(11), which means that the largest bin of $|[d_2]|$ may present at 1 even though the quantization step does not equal to 1, then wrong estimation occurs in such case. Therefore, we have to first apply an empirical measure to determine whether the quantization step is 1 or larger than 1. Based on our experiments, if

$$\frac{H(1)}{H(0)} > t \;\;\&\;\; H(1) > H_{max} \tag{12}$$

then the estimated quantization step $\hat{q}_{i,j} = 1$, where $H(k)$ is the number of values in $|[d_2(i,j)]|$ that having level k, and H_{max} denotes the maximum value in the set of $\{H(2), H(3), \ldots\}$. Please note that the threshold t should be larger than $8.5\%/91.5\% \approx 9.3\%$ according to Eq. (11). We have experimented with different t values ranging from 0.10 to 0.35 and found that the results are better when $t = 0.3$. Therefore, for all the experimental results presented in this paper, we set $t = 0.3$.

Otherwise, the quantization step can be obtained by

$$\hat{q}_{i,j} = \arg\min_{k}(\; k \;\mid\; H(k) = H_{max}, \quad k \geq 2) \tag{13}$$

C. Detecting Quantization Table

In this subsection, we will investigate the differences between the single JPEG image and its re-compressed versions using different quantization tables. And then propose an effective method for the detection of the quantization table.

As shown in Fig.1 and Eq. (9), the relationship between the de-quantized coefficients d_1' (single JPEG) and d_2' (double JPEG) can be written as follows.

$$d_2' = [\frac{d_2}{Q_2}] \times Q_2 = [\frac{d_1' + \varepsilon}{Q_2}] \times Q_2 \tag{14}$$

where ε is an approximate Gaussian distribution just as shown in Eq. (1), Q_2 is the quantization table used in the JPEG recompression. Please note that $d_1' \in Z$ is a multiple of the previous quantization table Q_1. In the following, we will analyze each frequency component before and after JPEG recompression, namely $d_1'(i,j)$ and $d_2'(i,j)$, where $0 \leq i,j \leq 7$, separately.

We let $Y' = d_1'(i,j) = k_1 q_1$, where $q_1 = Q_1(i,j)$ is the quantization step used in the first compression, $k_1 \in Z$ is an integer. $Y'' = d_2'(i,j)$ and $q_2 = Q_2(i,j)$ is the quantization step in JPEG recompression. Then Eq. (14) can be rewritten as follows.

$$Y'' = [\frac{Y' + \varepsilon}{q_2}] \times q_2 = [\frac{k_1 \times q_1 + \varepsilon}{q_2}] \times q_2 \tag{15}$$

According to the relationships between the quantization steps q_1 and q_2, we will investigate Eq. (15) in the following three cases:

- **case #1:** $q_2 = q_1$. (*e.g.* $Q_1 = Q_2$)

  $$\begin{aligned} Y'' &= [\tfrac{k_1 \times q_2 + \varepsilon}{q_2}] \times q_2 = [k_1 + \tfrac{\varepsilon}{q_2}] \times q_2 \\ &= k_1 \times q_1 + [\tfrac{\varepsilon}{q_2}] \times q_2 = Y' + [\tfrac{\varepsilon}{q_2}] \times q_2 \end{aligned}$$

  Thus we have

  $$P(Y'' = Y' | q_2 = q_1) = P([\frac{\varepsilon}{q_2}] = 0) \tag{16}$$

  Please note that events $[\frac{\varepsilon}{q_2}] = 0$ and $q_2 = q_1$ are independent.

- **case #2:** $q_2 \neq q_1, q_2 = 1$. (*e.g.* $QF_2 = 100$)

  $$Y'' = [\frac{k_1 \times q_1 + \varepsilon}{1}] \times 1 = Y' + [\varepsilon]$$

  Thus we have

  $$P(Y'' = Y' | q_2 \neq q_1, q_2 = 1) = P([\varepsilon] = 0) \tag{17}$$

  Please note that events $[\varepsilon] = 0$ and $q_2 \neq q_1, q_2 = 1$ are independent.

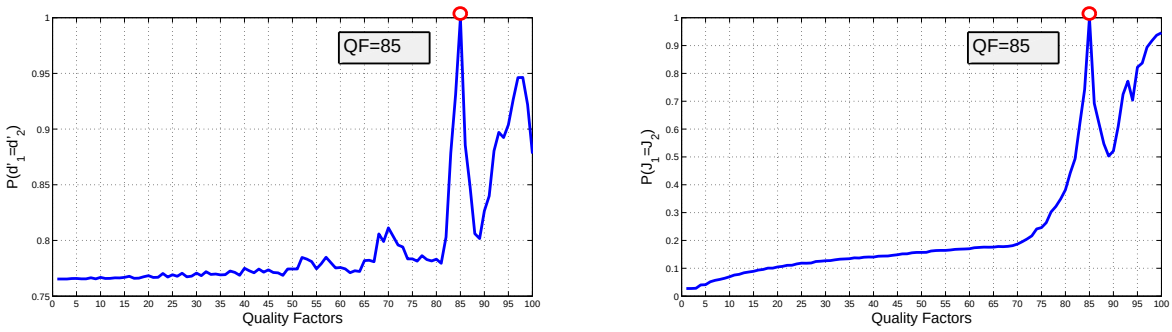
- **case #3:** $q_2 \neq q_1, q_2 > 1$. (*others*)
  - **subcase #3.1:** $q_2 \mid k_1 q_1$

    We let $Y' = k_1 q_1 = k_2 q_2$, where k_2 is an integer

    $$\begin{aligned} Y'' &= [\tfrac{k_1 \times q_1 + \varepsilon}{q_2}] \times q_2 = [\tfrac{k_2 \times q_2 + \varepsilon}{q_2}] \times q_2 \\ &= k_2 q_2 + [\tfrac{\varepsilon}{q_2}] \times q_2 = Y' + [\tfrac{\varepsilon}{q_2}] \times q_2 \end{aligned}$$

    Thus we have

    $$P(Y'' = Y' | q_2 \neq q_1, q_2 > 1, q_2 \mid k_1 q_1) = P([\frac{\varepsilon}{q_2}] = 0) \tag{18}$$



(a) Equal de-quantized coefficients (b) Equal pixel values

Fig. 4. The percentages of the equal de-quantized coefficients (a) and pixel values (b) as functions of the quality factors.

Please note that events $[\frac{\varepsilon}{q_2}] = 0$ and $q_2 \neq q_1, q_2 > 1, q_2 \mid k_1 q_1$ are independent.

– **subcase #3.2:** $q_2 \nmid k_1 q_1$

Assuming $Y'' = Y'$, we have $[\frac{k_1 q_1 + \varepsilon}{q_2}] \times q_2 = k_1 q_1$, thus $\frac{k_1 q_1}{q_2} = [\frac{k_1 q_1 + \varepsilon}{q_2}] \in Z$, then we get $q_2 | k_1 q_1$ contradiction. Thus we have

$$P(Y'' = Y' | q_2 \neq q_1, q_2 > 1, q_2 \nmid k_1 q_1) = 0 \tag{19}$$

From the Eq. (18),(19) and the law of total probability, we have that

$$\begin{aligned} & P(Y'' = Y' | q_2 \neq q_1, q_2 > 1) \\ = \; & P([\tfrac{\varepsilon}{q_2}] = 0) \times P(q_2 | k_1 q_1) + 0 \times P(q_2 \nmid k_1 q_1) \\ = \; & P([\tfrac{\varepsilon}{q_2}] = 0) \times P(q_2 | k_1 q_1) \end{aligned} \tag{20}$$

Based on Eq. (16), (17) and (20), we have that

$$P(Y'' = Y' | q_2 \neq q_1, q_2 = 1) \leq P(Y'' = Y' | q_2 = q_1) \tag{21}$$

$$P(Y'' = Y' | q_2 \neq q_1, q_2 > 1) \leq P(Y'' = Y' | q_2 = q_1) \tag{22}$$

The Eq. (21) and (22) indicate the de-quantized coefficients will be well preserved with the highest probability after JPEG recompression using the same quantization table ($Q_2 = Q_1$) compared with any others including the quantization table of ones ($QF_2 = 100$).

Here we give an example. Lena image with the size of 512×512 is JPEG compressed with a quality factor 85, followed by recompression with quality factors ranging from 1 to 100, respectively. Fig.4 (a) shows the percentages of the equal de-quantized coefficients before and after recompression (*i.e.*, $P(d'_1 = d'_2)$) with increasing the quality factors.

It is observed that the maximum value appears at the position 85, which is exactly the quality factor used in previous compression. Please note that for those quality factors less than 82, the quantization steps are not equal to the corresponding ones in the quantization table with $QF = 85$ and they are all larger than 1, thus **case #3** is considered. It is easy to check that $P(d_1'(i,j) = k_1q_1 = 0) = 76.54\%$ for the Lena image after single JPEG compressed with $QF = 85$, meaning that $P(k_1 = 0) = 76.54\%$. For those frequency components whose $k_1 = 0$, **subcase #3.1** is considered since $q_2 \mid k_1q_1 = 0$. In this subcase, most frequency coefficients will be preserved after recompression based on the Eq. (18), and that is the same probability as shown in **case #1** (see Eq. (16) and (18)). While for those frequency components whose $k_1 \neq 0$ (around 24%), $P(q_2|k_1q_1)$ is mostly equal to 0 especially when q_2 is large and $gcd(q_1, q_2) = 1$, where the function $gcd(a, b)$ denotes the greatest common divisor of a and b. Thus **subcase #3.2** is considered in these cases, and all the frequency coefficients will change after recompression based on the Eq. (19). Therefore the percentages for the QF less than 82 showing in Fig.4 (a) will fluctuate between 76% and 80%. Please note that some local maximum values occur around $QF = 70$ since that many quantization steps in $QF = 70$ are as twice as those for $QF = 85$, namely, $q_2 = 2q_1$. When k_1 is an even number, then we have $q_2 \mid k_1q_1$, which means that **case #3.1** is considered in such cases and thus more de-quantized coefficients will be preserved based on the Eq. (18). According to the relationship between Q_1 and Q_2 and the Eq. $(16) \sim (22)$, similar analysis can be obtained to explain the percentages with QFs $82 \sim 100$ shown in Fig.4 (a). These are exactly the results predicted by our theoretical analysis.

In practice, we can not obtain the de-quantized coefficients d_1' from a JPEG decompressed image J_1 because of the existence of the rounding errors $\Re$. Thus we investigate the difference between J_1 and J_2 instead of d_1' and d_2'. Please note that

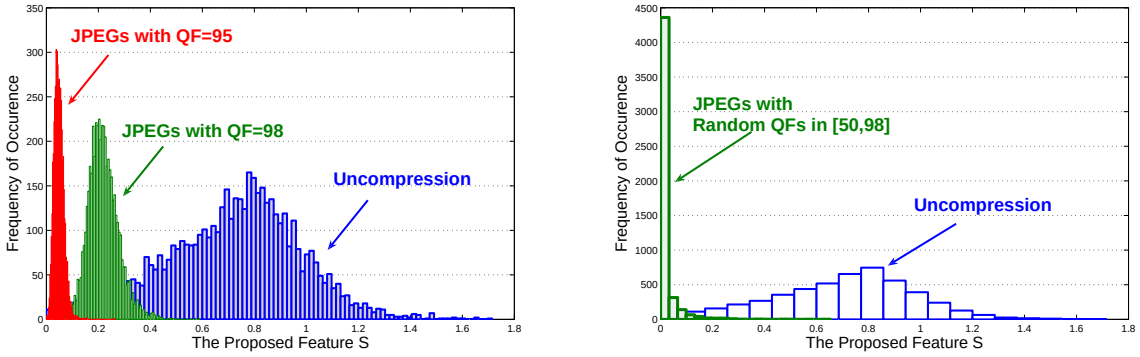
$$J_1 = [IDCT(d_1')], \quad J_2 = [IDCT(d_2')] \tag{23}$$

If all the frequency coefficients d_1' and d_2' within an 8×8 block are exactly the same, then the pixel values J_1 and J_2 in the spatial domain will be also preserved. Thus we define the similarity measure between two images J_1 and J_2 with size of $M \times N$ as follows.

$$R(J_1, J_2) = \frac{|E|}{MN} \tag{24}$$

where $E = \{(x, y)|J_1(x, y) = J_2(x, y), 1 \leq x \leq M, 1 \leq y \leq N\}$

Fig.4 (b) shows the feature R as a function of the quality factors. It's observed that the maximum value will also appear at the same position 85 as shown in Fig.4 (a). Based on this observation, we propose the following algorithm for detecting the quantization table. Let the JPEG decompressed image be J_1 and Q_1



(a) $QF = 98, 95$ (b) $QF \in \{50, \ldots 98\}$

Fig. 5. The histograms of the features s for original uncompressed images and their different JPEG compressed versions. The size of the test images used here is 32×32.

be the quantization table to be detected. We assume that the range of the quantization tables is known to us, *e.g.* the standard JPEG tables with quality factors ranging 1 to 100. We first recompress the image J_1 with all the possible quantization tables and obtain its decompressed versions $J_2(i)$, $i \in \{1, 2, \ldots 100\}$. Then the detected quality factor $\hat{QF}$ of J_1 can be obtained by

$$\hat{QF} = \arg\max_i (R(J_1, J_2(i)), i = 1, 2, \ldots 100) \tag{25}$$

III. EXPERIMENTAL RESULTS

In our experiments, we employ the Matlab JPEG Toolbox [33] for JPEG compression. We randomly select 1,000 images from each image dataset as used in Table I, namely Corel, NJIT, NRCS, SYSU and UCID, respectively. In all, there are 5,000 uncompressed color images. Those images are first converted into gray-scale images, which are then center-cropped into small blocks with sizes ranging from 256×256 to 8×8. The experimental results for JPEG history estimation, namely identifying JPEG images, estimating quantization steps and detecting quantization table are shown as follows.

A. Identifying JPEG Images

According to Eq. (7), we calculate the features s for the original uncompressed images and their different JPEG compressed versions and show the corresponding histograms in Fig.5. It can be observed from Fig.5 (a) that even with a slight compression, *e.g.* using the quality factors as high as 98, the features between uncompressed images and JPEG images are mostly separated. Usually, when the quality factors

TABLE II

AVERAGE ACCURACY (%) FOR JPEG IMAGE IDENTIFICATION. THE QUALITY FACTORS USED IN THE EXPERIMENTS ARE 98, 95, 85, 75 AND 50 SEPARATELY. THE THRESHOLDS ARE 0.2932, 0.3071, 0.3284, 0.3655, 0.3934, AND 0.3762 FOR THE SIX IMAGE SIZES. AND THE FPR (%) WE OBTAINED ARE 14.10, 11.58, 11.10, 12.52, 14.36 AND 21.06.

Quality Factor	256×256 Block	128×128 Block	64×64 Block	32×32 Block	16×16 Block	8×8 Block
QF=98	92.36	93.65	93.94	92.91	90.32	81.95
QF=95	92.95	94.21	94.45	93.74	92.79	89.32
QF=85	92.95	94.21	94.44	93.71	92.73	89.06
QF=75	92.95	94.21	94.42	93.69	92.68	88.75
QF=50	92.88	94.15	94.35	93.62	92.52	88.36

decrease (larger quantization steps are employed), the value of $\int_{R_1} p_{ac}(y)\,\mathrm{d}y$ in Eq. (7) will increase since more DCT coefficients would be quantized into zeros, which means that the lower the quality factors are, the smaller the features of the compressed images will be. Please compare the histograms of the features s for JPEG images with quality factors of 98 and 95 in Fig.5 (a). Therefore, thresholding the feature s can identify whether a bitmap image comes from a JPEG image or a uncompressed one.

To properly test the proposed feature s, we first use a minimum risk classification rule [35] to find a threshold t. For a given image size in the training stage, half of the uncompressed images and the corresponding JPEG compressed images with QF=98, the highest quality factor the proposed feature can detect reliably, are employed to obtain a proper threshold. These thresholds value are then used to identify the rest of JPEG images with QF=98, and all the other JPEG images with QF=95, 85, 75 and 50, respectively. The experimental results are shown in Table II. Here, we define the False Positive Rate (FPR) as the probability of the uncompressed images being wrongly determined as JPEG images, and thus it is fixed once the threshold is given for the same uncompressed image dataset. From Table II, we can observe that our method can achieve satisfactory accuracy of around 90% even the image size decreases to 8×8 and the quality factor is as high as 95, which shows that the proposed feature is very robust to the quality factors used previously as well as the image sizes.

To further show the effectiveness of the proposed feature, we compare our method with Fan's approach [14] and some typical blind measures for blocking artifacts of JPEG images used in [17], [18], [19], [20], [21], [22]. In this experiment, the quality factors are randomly selected in the range of $50\sim98$, the

TABLE III

AVERAGE ACCURACY (%) / FALSE POSITIVE RATE (%) FOR JPEG IMAGE IDENTIFICATION. THE VALUES WITH A ASTERISK (*) DENOTE THE BEST ACCURACY AND FPR AMONG THE SIX DETECTION METHODS. THE QUALITY FACTORS USED IN THE EXPERIMENTS ARE RANDOMLY SELECTED IN THE RANGE OF $[50, 98]$. IN THESE CASES, OUR BEST THRESHOLDS ARE $0.1112, 0.1138, 0.1468, 0.1416, 0.1232$ AND 0.1222 FOR THE SIX DIFFERENT IMAGE SIZES, RESPECTIVELY.

Detection Method	256×256 Block	128×128 Block	64×64 Block	32×32 Block	16×16 Block	8×8 Block
MSDSt [17]	53.44 / 35.68	54.00 / 51.80	53.32 / 54.60	53.56 / 49.48	-	-
Wang's [18]	77.48 / 15.76	73.22 / 19.68	69.58 / 28.92	64.48 / 45.88	62.62 / 48.80	-
Wang's [19]	73.94 / 25.48	57.40 / 39.40	53.42 / 43.96	51.10 / 61.64	-	-
Liu's [20]	76.50 / 3.44	66.20 / 16.12	62.50 / 14.36	59.50 / 28.80	56.94 / 44.00	-
Fan's [14]	94.50 / 3.24	86.38 / 12.44	74.14 / 28.04	61.28 / 49.48	50.58 / 88.28	-
Proposed	**98.64 / 1.00 ***	**98.52 / 1.08 ***	**97.96 / 2.72 ***	**97.62 / 2.84 ***	**96.24 / 3.24 ***	**95.08 / 5.80 ***

commonly used range for lossy JPEG compression. The histograms of our features are illustrated in Fig.5 (b), it is also expected that the proposed method can obtain good results since most features of the uncompressed and JPEG compressed images are clearly separated with a proper threshold.

Table III shows the average accuracies and false positive rates for the six detection methods. It is observed that the proposed method outperforms the existing methods significantly. Even the image size is as small as 8×8, our method can still obtain the average accuracy of 95.08% and FPR of 5.8%. While the performance of existing methods is very sensitive to image sizes, it usually drops significantly and even becomes random guessing with decreasing the image size, *e.g.* less than 32×32, and all of them will fail to detect image blocks whose sizes are less then 9×9. One of the reasons is that these methods tried to measure the artificial discontinuities/stpes between the neighboring 8×8 blocks introduced by the lossy JPEG compression. In such cases, the image content edges may be located at the boundaries between the blocks and thus confuse the strength of the blocking artifacts. Therefore, the existing methods usually need sufficient statistics (*e.g.* larger than 256×256) to eliminate the effects of content edges, especially for those JPEG images with slight compression, *e.g* $QF > 90$. Quite different from the existing methods, our method aims to detect the quantization artifacts within every single 8×8 block rather than the discontinuities between the blocks. And therefore the proposed feature can eliminate the effect of the content edges very effectively.

B. Estimating Quantization Steps

If a bitmap image is identified as JPEG decompressed image, we can further estimate the quantization steps for each frequency component (63 AC components in all) according to the Eqs. (12) and (13). The experimental results evaluated on 5,000 images of 256×256 are shown below. The corresponding quantization steps for the 63 AC components can be obtained according to Eq. (26) in the Appendix.

$$\begin{pmatrix} - & \underline{46.40} & 68.42 & 89.82 & 92.74 & 98.90 & 99.62 & 99.62 \\ \underline{31.70} & 65.98 & 75.68 & 91.92 & 97.14 & 99.54 & 99.66 & 99.54 \\ 64.98 & 73.36 & 90.72 & 93.28 & 98.50 & 99.62 & 99.56 & 99.56 \\ 73.86 & 91.04 & 93.08 & 97.56 & 99.46 & 99.54 & 99.36 & 99.38 \\ 91.54 & 93.54 & 98.58 & 99.68 & 99.62 & 98.84 & 98.50 & 98.52 \\ 94.36 & 98.88 & 99.86 & 99.78 & 99.64 & 98.48 & 96.92 & 97.02 \\ 99.74 & 99.90 & 99.68 & 99.34 & 98.48 & 96.58 & 94.52 & 94.94 \\ 99.76 & 99.24 & 98.74 & 98.34 & 96.84 & 96.44 & 95.54 & 93.30 \end{pmatrix}$$

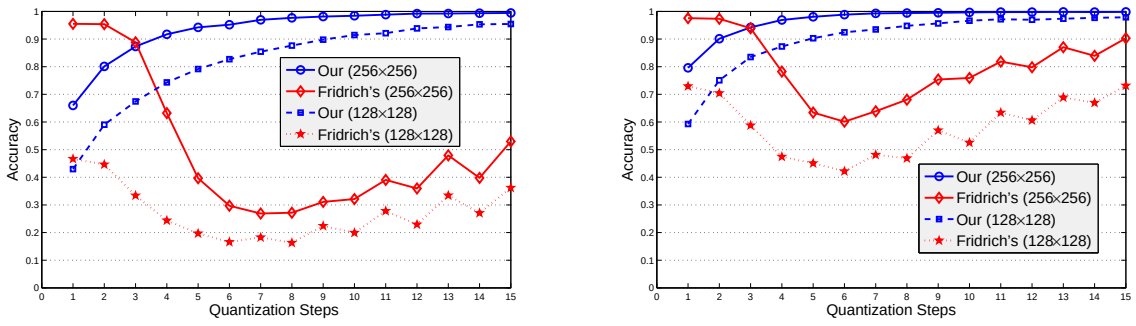
(a). Average accuracy (%) for $QF = 95$

$$\begin{pmatrix} - & 95.48 & 98.64 & 99.02 & 98.58 & 81.98 & 89.86 & 71.92 \\ 96.84 & 99.06 & 99.80 & 99.80 & 98.98 & 88.62 & 77.52 & 66.00 \\ 99.22 & 99.52 & 99.80 & 99.42 & 96.32 & 82.64 & 59.20 & 54.94 \\ 96.48 & 99.78 & 99.58 & 98.36 & 88.54 & \underline{46.38} & \underline{34.80} & \underline{37.70} \\ 99.62 & 99.44 & 96.70 & 86.38 & 66.78 & \underline{13.54} & \underline{8.24} & \underline{13.00} \\ 87.84 & 96.86 & 86.28 & 70.86 & \underline{37.94} & \underline{7.30} & \underline{2.68} & \underline{3.86} \\ 92.12 & 76.70 & 52.82 & \underline{28.24} & \underline{8.20} & \underline{1.30} & \underline{0.72} & \underline{0.94} \\ 56.48 & \underline{29.14} & \underline{15.88} & \underline{8.38} & \underline{2.06} & \underline{2.16} & \underline{0.88} & \underline{0.76} \end{pmatrix}$$

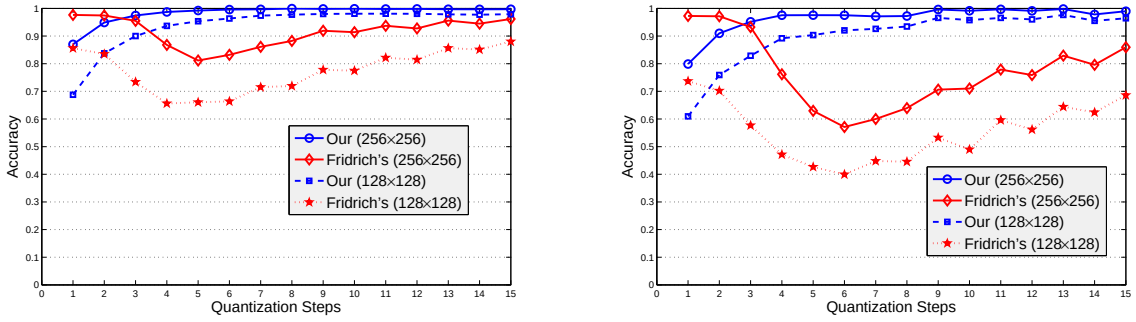
(b). Average accuracy (%) for $QF = 50$

We underlined the detection accuracies that are less than 50%. It is observed that for the higher quality factors, *e.g.* $QF = 95$, the proposed method performs better on estimating the quantization steps for the median and high frequency components than those for the low ones. The reason is that most quantization steps for the low frequency components are close to 0, namely, 1 or 2. Since the DCT coefficients d_2 with values 0 or 1 (1 or 2) will interact due to the error ε (please refer to Eqs. (9) $\sim$ (11)), which makes that the location of the largest histogram bin may not located at the quantization step but its neighbors. In such cases, our method would fail. For the lower quality factors, *e.g.* $QF = 50$, most higher frequency coefficients of natural images will be quantized to zeros due to the large quantization steps (*e.g.* for those quantization steps larger than 70), and thus the detection accuracy would inevitably decrease significantly due to insufficient number of effective data samples for the estimation.

To show the effectiveness of the proposed method, we compare it with the Fridrich's method [15]. Firstly, we evaluate the performances on a set of particular quantization steps in the four AC components AC(1,1), AC(2,2), AC(3,3) and AC(4,0). In the experiments, we select the proper quality factors whose



(a) AC(1,1) (b) AC(2,2)



(c) AC(3,3) (d) AC(4,0)

Fig. 6. The detection accuracy as a function of the quantization steps ranging from 1 to 15. Four AC components AC(1, 1), AC(2, 2), AC(3, 3) and AC(4, 0) are investigated with some proper quality factors.

corresponding quantization steps are from 1 to 15. Taking AC(1,1) for example, the selected quality factors are 95, 90, 86, 82, 78, 73, 69, 65, 61, 57, 53, 49, 45, 42 and 39. Fig. 6 shows the average accuracy as a function of the quantization steps. It is observed that our accuracy usually increases with increasing the quantization step, and outperforms that of method [15] in most situations.

Secondly, we evaluate the overall performances of the two methods on the whole quantization table. Five quantization tables with quality factors 95, 90, 85, 75 and 50 are investigated in the experiments. Table IV shows the average accuracies for the test images with different sizes and quality factors. It is also observed that our method outperforms the method [15] in most cases, especially for those images with smaller sizes. For instance, our average accuracy is 81.97% for the images with size of 128×128 and with the quality factor 85, that is around 16% improvement over the method [15].

TABLE IV

AVERAGE ACCURACY FOR QUANTIZATION STEPS ESTIMATION. THE VALUE WITH A ASTERISK (*) DENOTES THE BETTER ACCURACY BETWEEN THE TWO DETECTION METHODS.

Quality Factor	Detection Method	256×256 Block	128×128 Block	64 × 64 Block
QF=95	Proposed	92.93	**85.22 ***	**70.21 ***
	Method [15]	**93.42 ***	75.31	49.07
QF=90	Proposed	**93.70 ***	**85.45 ***	**68.30 ***
	Method [15]	89.48	71.04	49.27
QF=85	Proposed	**91.44 ***	**81.97 ***	**63.56 ***
	Method [15]	82.75	66.42	46.77
QF=75	Proposed	**82.35 ***	**72.47 ***	**54.49 ***
	Method [15]	72.45	58.49	40.55
QF=50	Proposed	**62.05 ***	**54.71 ***	**41.15 ***
	Method [15]	57.55	45.15	31.25

C. Detecting Quantization Table

As shown in the previous subsection III-B, it is difficult to estimate those larger quantization steps for the high frequency components of natural images when the quality factor is low (*e.g.* QF=50) and the size of image is small (*e.g.* 128×128). In such cases, we require some prior knowledge to determine those uncertain quantization steps. We assume that we can obtain the range of the quantization tables in advance. And this is a reasonable assumption in some forensics applications. For instance, we wish to determine whether or not a given bitmap image is coming from some questionable digital cameras or has been JPEG compressed with some photo-editing software packages. In this subsection, we will present some experimental evidences to show that the detection will become more reliable in such cases. The algorithm is that for a given JPEG decompressed image J_1, we recompress it with all the candidate tables and obtain the corresponding recompressed versions in the spatial domain, and then find the closest one to the image J_1, say J_2, according to the Eq. (24). Finally, the quantization table used in J_2 is the detected table for the questionable image J_1. The validity of the detection algorithm was proved by the Eq. (21) and (22).

In this paper, we assume the candidate quantization tables are the standard JPEG tables with quality factors ranging from 1 to 100. Thus the Eq. (25) is employed for the quality factor estimation. Please note that our method is also effective for the images with non-standard tables such as the quantization

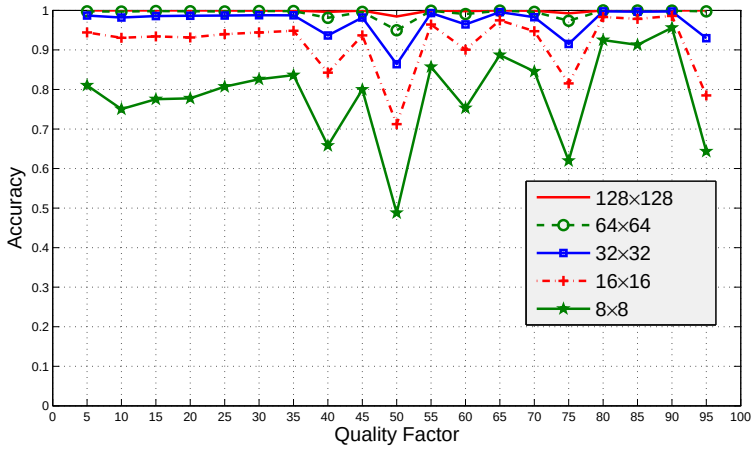


Fig. 7. Detection accuracy as a function of the quality factors.

TABLE V

THE AVERAGE ACCURACIES FOR DIFFERENT QUALITY FACTORS USING THE TWO METHODS.

Image Size	Detection Method	QF=10	QF=20	QF=30	QF=40	QF=50	QF=60	QF=70	QF=80	QF=90
64×64	Method [16]	4.82	17.52	10.90	17.54	0	29.20	32.58	69.80	69.66
	Proposed	**99.60 ***	**99.70 ***	**99.66 ***	**98.10 ***	**94.52 ***	**98.92 ***	**99.42 ***	**99.88 ***	**99.68 ***
128×128	Method [16]	13.08	42.74	34.38	35.18	0	55.00	43.96	89.64	84.24
	Proposed	**99.96 ***	**99.96 ***	**99.94 ***	**99.64 ***	**98.52 ***	**99.86 ***	**99.96 ***	**99.98 ***	**99.92 ***

tables used in Photoshop, GIMP and various digital cameras [13], as long as the range of the tables is known. In the experiments, the original uncompressed images with sizes from 8×8 to 128×128 are first JPEG compressed using quality factors ranging from 5 to 95 with a step size of 5 to obtain the test JPEG images. Fig.7 shows the average accuracy evaluated on the test images in different cases. Please note that the accuracy is relatively lower for detecting the JPEG images with $QF = 50$. The reason is that the first 19 quantization steps (along the zigzag scanning order) in the quantization tables with quality factors 49, 50 and 51 are exactly the same. It is well known that most energy in natural images is concentrated on *DC* and low-frequency components, while most higher frequency components would be quantized to zeros after severe JPEG compression, *e.g.* the QF is around 50. Therefore more effective data samples (larger image sizes) are needed to distinguish them. It is observed that the detection accuracy would

increase with increasing image sizes. When the image size becomes larger than 64×64, the average accuracy can achieve over 94.52% for all the cases.

To further show the effectiveness of the proposed method, we compare our method with the Benford's law (first digit law) based method [16] and show the experimental results in Table V. Obviously, the proposed method outperforms the existing method [16] significantly.

IV. CONCLUDING REMARKS

This paper introduces JPEG error analysis to the study of image forensics. Through analyzing the various errors of JPEG image compression and their relationships in double JPEG compression, we have developed three image forensic analysis schemes. A simple and very effective 1-D feature has been successfully introduced to identify if a given bitmap image has previously been JPEG compressed. Once a bitmap is identified as JPEG image, we have developed techniques to further estimate the quantization steps and detect quantization table used for the JPEG image. We have shown that the new methods worked effectively and significantly outperformed existing techniques in the literature. Please note that we just present theoretical analysis and experimental results for gray-scale images in this paper, while the proposed methods can be extended to detect color images in a similar way.

APPENDIX

In the appendix, we will present the relationship between the quality factors and the quantization tables used in the paper. It's well known that quality factor is the scaling factor for a default table, and it is widely used in most JPEG compression schemes. Usually, the default quantization table t is defined as:

$$t = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}$$

Then the quantization tables with a quality factor can be obtain by

$$Table_{(QF)} = \begin{cases} \lfloor t \times \dfrac{50}{QF} + \dfrac{1}{2} \rfloor & 1 \leq QF < 50 \\ \lfloor t \times (2 - \dfrac{QF}{50}) + \dfrac{1}{2} \rfloor & 50 \leq QF \leq 100 \end{cases} \tag{26}$$

where $QF \in \{1, 2, \ldots 100\}$ denotes the quality factors, $y = \lfloor x \rfloor$ denotes the floor function which maps a real number x to the next smallest integer y, and if y is less than 1, let $y = 1$.

Therefore, the readers can obtain the corresponding quantization tables for a given quality factor QF. For instance, the quantization table with QF=50 is the default table t as shown above, and the tables with QF=98, 95 and 90 are as follows:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 3 & 2 \\ 1 & 1 & 1 & 1 & 2 & 3 & 3 & 2 \\ 1 & 1 & 1 & 2 & 3 & 4 & 4 & 3 \\ 1 & 1 & 2 & 3 & 3 & 4 & 5 & 4 \\ 2 & 3 & 3 & 3 & 4 & 5 & 5 & 4 \\ 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \end{pmatrix}$$

Quantization Table with QF=98

$$\begin{pmatrix} 2 & 1 & 1 & 2 & 2 & 4 & 5 & 6 \\ 1 & 1 & 1 & 2 & 3 & 6 & 6 & 6 \\ 1 & 1 & 2 & 2 & 4 & 6 & 7 & 6 \\ 1 & 2 & 2 & 3 & 5 & 9 & 8 & 6 \\ 2 & 2 & 4 & 6 & 7 & 11 & 10 & 8 \\ 2 & 4 & 6 & 6 & 8 & 10 & 11 & 9 \\ 5 & 6 & 8 & 9 & 10 & 12 & 12 & 10 \\ 7 & 9 & 10 & 10 & 11 & 10 & 10 & 10 \end{pmatrix}$$

Quantization Table with QF=95

$$\begin{pmatrix} 3 & 2 & 2 & 3 & 5 & 8 & 10 & 12 \\ 2 & 2 & 3 & 4 & 5 & 12 & 12 & 11 \\ 3 & 3 & 3 & 5 & 8 & 11 & 14 & 11 \\ 3 & 3 & 4 & 6 & 10 & 17 & 16 & 12 \\ 4 & 4 & 7 & 11 & 14 & 22 & 21 & 15 \\ 5 & 7 & 11 & 13 & 16 & 21 & 23 & 18 \\ 10 & 13 & 16 & 17 & 21 & 24 & 24 & 20 \\ 14 & 18 & 19 & 20 & 22 & 20 & 21 & 20 \end{pmatrix}$$

Quantization Table with QF=90

ACKNOWLEDGMENT

The authors would like to thank Dr. Dongdong Fu at New Jersey Institute of Technology, New Jersey, USA, and YuanGen Wang at Sun Yat-Sen Univ. Guangzhou, China, for their helpful discussions and thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] B. Zhu, M.D. Swanson, and A.H. Tewfik, "When seeing isn't believing [multimedia authentication technologies]," *IEEE Signal Processing Magazine*, vol. 21, no. 2, pp. 40–49, Mar. 2004.

[2] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1079–1107, Jul. 1999.

[3] A. Swaminathan, Y. Mao, and M. Wu, "Robust and secure image hashing," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 215–230, Jun. 2006.

[4] H. Farid, "A survey of image forgery detection," *IEEE Signal Processing Magazine*, vol. 2, no. 26, pp. 16–25, Mar. 2009.

[5] W. Luo, Z. Qu, F. Pan, and J. Huang, "A survey of passive technology for digital image forensics," *Frontiers of Computer Science in China*, vol. 1, no. 2, pp. 1673–7350, May 2007.

[6] J. Lukáš and J. Fridrich, "Estimation of primary quantization matrix in double compressed JPEG images," in *Proceedings of Digital Forensic Research Workshop*, Cleveland, OH, USA, Aug. 5-8 2003.

[7] A.C. Popescu, *Statistical Tools for Digital Image Forensics*, Ph.D. thesis, Department of Computer Science, Dartmouth College, Hanover, NH, USA, 2005.

[8] W. Luo, Z. Qu, J. Huang, and G. Qiu, "A novel method for detecting cropped and recompressed image block," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Honolulu, Hawaii, USA, Apr. 2007, vol. 2, pp. 217–220.

[9] H. Farid, "Exposing digital forgeries from JPEG ghosts," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 1, pp. 154–160, Mar. 2009.

[10] S. Ye, Q. Sun, and E. Chang, "Detecting digital image forgeries by measuring inconsistencies of blocking artifact," in *Proceedings of IEEE International Conference on Multimedia and Expo*, July 2007, pp. 12–15.

[11] K. Solanki, A. Sarkar, and B.S. Manjunath, "YASS: Yet another steganographic scheme that resists blind steganalysis," in *Proceedings of 9th International Workshop on Information Hiding*, Saint Malo, France, 2007, vol. 4567, pp. 16–31.

[12] B. Li, Y.Q. Shi, and J. Huang, "Steganalysis of YASS," in *Proceedings of the 10th ACM workshop on Multimedia and security*, Oxford, UK, Sep. 2008, pp. 139–148.

[13] H. Farid, "Digital image ballistics from JPEG quantization," Tech. Rep. TR2006-583, Department of Computer Science, Dartmouth College, 2006.

[14] Z. Fan and R. de Queiroz, "Identification of bitmap compression history: JPEG detection and quantizer estimation," *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 230–235, Feb. 2003.

[15] J. Fridrich, M. Goljan, and R. Du, "Steganalysis based on JPEG compatibility," in *Special session on Theoretical and Practical Issues in Digital Watermarking and Data Hiding, Multimedia Systems and Applications IV*, Denver, Co, USA, Aug. 2001, pp. 275–280.

[16] D. Fu, Y.Q. Shi, and W. Su, "A generalized benford's law for JPEG coefficients and its applications in image forensics," in *Proceedings of SPIE Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents*, San Jose, CA, USA, 2007, vol. 6505, p. 65051L.

[17] G.A. Triantafyllidis, D. Tzovaras, and M.G. Strintzis, "Blocking artifact detection and reduction in compressed data," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 10, pp. 877–890, Oct. 2002.

[18] Z. Wang, H.R. Sheikh, and A.C. Bovik, "No-reference perceptual quality assessment of JPEG compressed images," in *Proceedings of International Conference on Image Processing*, 2002, vol. 1, pp. 477–480.

[19] Z. Wang, A.C. Bovik, and B.L. Evan, "Blind measurement of blocking artifacts in images," in *Proceedings of International Conference on Image Processing*, 2000, vol. 3, pp. 981–984.

[20] S. Liu and A.C.Bovik, "Efficient DCT-domain blind measurement and reduction of blocking artifacts," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 12, pp. 1139–1149, Dec. 2002.

[21] S. Minami and A. Zakhor, "An optimization approach for removing blocking effects in transform coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 2, pp. 74–82, Apr. 1995.

[22] G. Lakhani and N. Zhong, "Derivation of prediction equations for blocking effect reduction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 3, pp. 415–418, Apr. 1999.

[23] A. Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 2, no. 1, pp. 91–95, Mar. 1992.

[24] A.Liew and H. Yan, "Blocking artifacts suppression in block-coded images using overcomplete wavelet representation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 4, pp. 450–461, Apr. 2004.

[25] J. Luo, C. Chen, K.J. Parker, and T.S. Huang, "Artifact reduction in low bit rate DCT-based image compression," *IEEE Transactions on Image Processing*, vol. 5, no. 9, pp. 1363–1368, Sep. 1996.

[26] B. Li, Y.Q. Shi, and J. Huang, "Detecting doubly compressed JPEG images by using mode based first digit features," in *IEEE Workshop on Multimedia Signal Processing,*, Oct. 2008, pp. 730–735.

[27] C. Chen, Y.Q. Shi, and W. Su, "A machine learning based scheme for double JPEG compression detection," in *Proceedings of International Conference on Pattern Recognition*, Dec. 2008, pp. 1–4.

[28] T. Pevny and J. Fridrich, "Detection of double-compression in JPEG images for applications in steganography," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 2, pp. 247–258, Jun. 2008.

[29] T. Pevny and J. Fridrich, "Estimation of primary quantization matrix for steganalysis of double-compressed JPEG images," in *Proceedings of SPIE Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents*. 2008, vol. 6819, p. 681911, SPIE.

[30] B. Mahdian and S. Saic., "Detecting double compressed JPEG images," in *Submitted to the 3rd International Conference on Imaging for Crime Detection and Prevention*, 2009.

[31] [online] Available at:, "NRCS photo gallery," http://photogallery.nrcs.usda.gov/, 2005.

[32] G. Schaefer and M. Stich, "UCID: an uncompressed color image database," in *Proceedings of SPIE on Storage and Retrieval Methods and Applications for Multimedia*, San Jose, CA, USA, 2003, vol. 5307, pp. 472–480.

[33] [online] Available at:, ," http://www.philsallee.com/jpegtbx/index.html.

[34] R. Reininger and J. Gibson, "Distributions of the two-dimensional DCT coefficients for images," *IEEE Transactions on Communications*, vol. 31, no. 6, pp. 835–839, Jun. 1983.

[35] S. Theodoridis and K. Koutroumbas, *Pattern recognition (second edition)*, China Machine Press, 2003.