Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Estimating JPEG compression history of bitmaps based on factor histogram

Jianquan Yang [a], Guopu Zhu [a,*], Jiwu Huang [b], Xi Zhao [c]

[a] *Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, GD 518055, PR China*
[b] *College of Information Engineering, Shenzhen University, Shenzhen, GD 518060, PR China*
[c] *School of Computer Science and Information Engineering, Tianjin University of Science and Technology, Tianjin 300222, PR China*

ARTICLE INFO

Article history:
Available online 2 April 2015

Keywords:
Image forensics
Compression history
JPEG compression
Bitmap

ABSTRACT

Estimation of JPEG compression history for bitmaps has drawn increasing attention in the past decade due to its extensive applications in image processing, image forensics and steganalysis. In this paper, we propose a novel statistic named factor histogram for estimating the JPEG compression history of bitmaps. In a statistical sense, the factor histogram decreases with the increase of its bin index for uncompressed bitmaps. Whereas, it exhibits a local maximum at the bin index corresponding to the quantization step for JPEG decompressed bitmaps, which makes itself no longer decrease. Based on these characteristics, we propose to identify decompressed bitmaps by measuring the monotonicity of factor histogram, and to estimate the quantization step of each frequency by locating the bin index of the local maximum in factor histogram. Experimental results demonstrate that the proposed method outperforms the existing methods for a range of image sizes, meanwhile maintaining low computational cost.

1. Introduction

Joint Photographic Experts Group (JPEG) lossy compression is one of the most popular image compression standards. In some cases, a JPEG image could be decompressed and saved into bitmap form such as Windows bitmap (BMP) or Tagged Image File Format (TIFF). As a result, the JPEG header information is discarded, so its corresponding quantization step matrix is untraceable. The research of retrieving the unknown JPEG compression history (i.e., identifying whether a bitmap is decompressed from a JPEG image or not, and estimating the used quantization step matrix of a decompressed bitmap) has drawn considerable attention, as it plays a crucial role in many scenarios of image processing and digital forensics [1]. For instance, many de-blocking algorithms require the knowledge of compression history in order to guide the process of removing blocking artifacts from a decompressed bitmap [2,3]. With the knowledge of JPEG compression history, one can recompress a decompressed bitmap in a way that avoids further distortion while simultaneously achieving a small file-size [4,5]. In the steganalysis field, analyzers would be able to deal with the effect of JPEG compression to improve the accuracy of detecting hiding data [6,7] if the compression history of the stego bitmap is known. Furthermore, the tampered regions in a forged image can be localized by first estimating the compression history block by block, and then revealing the inconsistence of the compression history among these blocks [8,9].

The estimation of JPEG compression history for bitmaps has been widely studied [2,3,6,8–13] in the past decade. In the aspect of decompressed bitmap identification, Fan et al. [10] proposed to measure the strength of blocking artifacts in pixel domain and then to make a two-class decision by thresholding the strength metric. Fu et al. [11] presented the generalized Benford's law for modeling the distribution of the first digits of Discrete Cosine Transform (DCT) coefficients, and designed a support vector machine (SVM) based classifier to detect the presence of JPEG compression. Recently, Luo et al. [12] found that the number of DCT coefficients in the range $(-1, 1)$ is much greater than that in the range $(-2, -1) \cup (1, 2)$ for decompressed bitmaps. They then took the ratio of number of coefficients between the two ranges as a distinguished feature to determine whether a given bitmap has been previously JPEG compressed. There are also several works [2,3] that could be extended to identify decompressed bitmaps. However, the performance of these works is far from satisfactory, as reported in [12].

In the aspect of quantization step estimation for bitmaps, Fridrich et al. [6] proposed to estimate the quantization step of each frequency by performing a so-called "compatibility test" over all possible steps. Fan et al. [10] designed a complicated like-

[*] Corresponding author.
E-mail addresses: jq.yang@siat.ac.cn (J. Yang), guopu.zhu@gmail.com (G. Zhu), jwhuang@szu.edu.cn (J. Huang), xi.zhao@tust.edu.cn (X. Zhao).

http://dx.doi.org/10.1016/j.dsp.2015.03.014
1051-2004/© 2015 Elsevier Inc. All rights reserved.

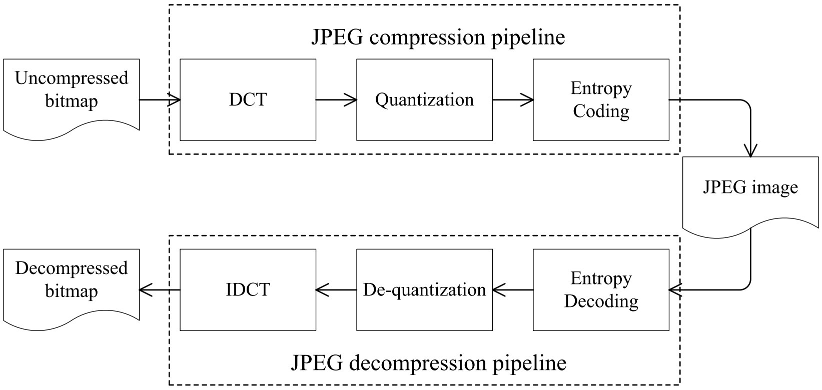


Fig. 1. Generation of JPEG decompressed bitmap.

lihood function and then selected a candidate quantization step which could maximize the likelihood function as the step estimate. Based on Fan et al.'s method, Neelamani et al. [13] developed a dictionary-based and a lattice-based methods to estimate quantization step for decompressed color bitmaps. A common feature of these methods [6,10,13] is that they require a time-consuming search over all possible candidate steps [8], which may result in these algorithms not meeting the requirement of real-time applications. In order to accelerate the process of quantization step estimation, Ye et al. [8] developed a step estimation method by counting the local maxima in the power spectrum of DCT coefficient histogram. Luo et al. [12] proposed to estimate the quantization step by locating the maximum bin in the DCT coefficient histogram with several empirical rules. Recently, Lin et al. [9] further improved Ye et al.'s method by classifying the patterns of individual frequencies into four classes, and then extracting the quantization step adaptively from the energy density spectrum (EDS) of DCT coefficient histogram and the Fourier transform of EDS. In addition, there are also some works aiming to detect doubly JPEG compressed images and estimate quantization parameters of the primary JPEG compression [14–21]. These works typically assumed that the quantized DCT coefficients can be directly decoded from the input JPEG file, or the quantization table of the last (i.e., secondary) JPEG compression is known.

In this paper, we focus on estimating the JPEG compression history for bitmaps. The main contributions of the paper are as follows:

1. A novel statistic named factor histogram is proposed by studying the distribution of the integer factors of DCT coefficients. A theoretical analysis is provided on the characteristics of the factor histogram.
2. An effective factor histogram-based method is developed to estimate the JPEG compression history of bitmaps. The presence of JPEG compression is detected by measuring the monotonicity of factor histogram, and the quantization step is estimated by selecting a histogram bin with the maximum index from those bins larger than a threshold.

Extensive experimental results demonstrate that our proposed method outperforms the existing methods for a range of image sizes, meanwhile maintaining low computational cost.

The rest of this paper is organized as follows. Section 2 introduces the concept of factor histogram and investigates its characteristics. Section 3 presents the details of our proposed method for estimating the JPEG compression history of bitmaps. Experimental results are shown and discussed in Section 4. Finally, conclusions are drawn in Section 5.

2. Factor histogram

As shown in Fig. 1, a decompressed bitmap is generated by compressing an uncompressed bitmap into a JPEG image and then decompressing it back into bitmap form. During the JPEG compression stage, three main operations, which include DCT, quantization and entropy coding, are performed sequentially; During the decompression stage, entropy decoding, de-quantization and inverse DCT (IDCT) are performed. Since DCT and IDCT, as well as entropy coding and decoding, is a pair of lossless operations, we therefore concentrate on quantization and de-quantization, which mainly cause the JPEG compression artifacts. By modeling the operations of quantization and de-quantization, the concept of factor histogram and its characteristics are presented as follows.

2.1. Definition of factor histogram

Let $\{a_n\}_{n=1}^N$ denote a real sequence with length N. $\{a_n\}_{n=1}^N$ is quantized with step q and then de-quantized to generate an integer sequence $\{b_n\}_{n=1}^N$, each element of which can be formulated by

$$b_n = [a_n/q] \times q \tag{1}$$

where $[\cdot]$ denotes the round function, and q is a positive integer. According to Eq. (1), q is a positive integer factor ("factor" for short) of b_n, we therefore concern the distribution of the factors of elements in $\{b_n\}_{n=1}^N$, which could provide us with an alternative way to study the quantization effect. We first define the "factor set" of an integer v as

$$F(v) \triangleq \{u | u \in \mathbb{N}, \mathrm{mod}(v,u) = 0\} \tag{2}$$

where $\mathrm{mod}(v,u)$ denotes "v modular u". Obviously, we have $F(0) = \mathbb{N}$.

We then calculate the factor set for each element of $\{b_n\}_{n=1}^N$ and count the number of factor sets containing a specific factor x, which can be expressed as

$$h_f(x) \triangleq \sum_{n=1}^{N} g\Big(x, F(b_n)\Big) \tag{3}$$

where $g(x, S)$ denotes the function that takes 1 if $x \in S$ and 0 otherwise. We call $h_f(x)$ as the factor histogram of $\{b_n\}_{n=1}^N$. Note that $h_f(1) \equiv N$, because 1 is the positive factor of any integer.

Furthermore, we analyze the fast calculation of factor histogram. Given an integer sequence $\{b_n\}_{n=1}^N$, we could achieve its factor histogram by factorizing its elements one by one according to Eq. (3). If $\{b_n\}_{n=1}^N$ contains reduplicate elements, the computational efficiency could be improved. The idea is that we could count the repeating time of each different element beforehand in order to avoid factorizing a reduplicate element repeatedly. To achieve this, we first calculate the histogram of the integer sequence $\{b_n\}_{n=1}^N$, and then obtain the factor histogram by

$$h_f(x) = \sum_{y=y_{min}}^{y_{max}} h(y) g\Big(x, F(y)\Big) \tag{4}$$

where y_{min} (y_{max}) denotes the minimum (maximum) element of $\{b_n\}_{n=1}^N$, and $h(y)$ denotes the histogram of $\{b_n\}_{n=1}^N$, which is given by

$$h(y) \triangleq \sum_{n=1}^{N} \delta(y - b_n),\ y \in \mathbb{Z}$$

Eq. (4) not only provides an accelerated approach to calculate factor histogram, but it also reveals the relationship between the factor histogram of $\{b_n\}_{n=1}^N$ and the histogram of $\{b_n\}_{n=1}^N$.

Finally, the factor histogram is normalized into the range [0, 1] by

$$\bar{h}_f(x) = h_f(x)/N,\ N > 0 \tag{5}$$

Note that $\bar{h}_f(1) \equiv 1$, which means that bin 1 is an intrinsic maximum bin of the normalized factor histogram $\bar{h}_f(x)$.

2.2. Characteristics of factor histogram

In the literature, it has been widely studied that the distribution of block-wise DCT coefficients from uncompressed bitmaps could be modeled as Laplacian, Gaussian, generalized Gaussian, as well as sum of Gaussians [22]. Let $p_{ac}(u;\boldsymbol{\theta})$ denote the probability density function (PDF) of an AC frequency coefficient, where $\boldsymbol{\theta}$ denotes the PDF's parameter(s). Since $\boldsymbol{\theta}$ is not related to the following derivation, we omit $\boldsymbol{\theta}$ for simplicity. For uncompressed bitmaps, the PDF is assumed to satisfy $p_{ac}(u) = p_{ac}(-u)$ and $p_{ac}(u_1) \geq p_{ac}(u_2)$ if $|u_1| \leq |u_2|$. According to Eq. (4), the normalized factor histogram can be represented as

$$\begin{aligned}
\bar{h}_f(x) &= \sum_{y=-\infty}^{\infty} \bar{h}(y) g\Big(x, F(y)\Big) \\
&= \sum_{y=-\infty}^{\infty} \int_{y-0.5}^{y+0.5} p_{ac}(u)du \cdot g\Big(x, F(y)\Big) \\
&= \sum_{n=-\infty}^{\infty} \int_{nx-0.5}^{nx+0.5} p_{ac}(u)du \cdot g\Big(x, F(nx)\Big) \\
&= \sum_{n=-\infty}^{\infty} \int_{nx-0.5}^{nx+0.5} p_{ac}(u)du
\end{aligned}$$

Let $0 < x_1 < x_2$ ($x_1,\ x_2 \in \mathbb{N}$). We then have

$$\begin{aligned}
&\bar{h}_f(x_1) - \bar{h}_f(x_2) \\
&= 2\sum_{n=1}^{\infty} \left(\int_{nx_1-0.5}^{nx_1+0.5} p_{ac}(u)du - \int_{nx_2-0.5}^{nx_2+0.5} p_{ac}(u)du \right) \\
&\geq 2\sum_{n=1}^{\infty} \Big[p_{ac}(nx_1 + 0.5) - p_{ac}(nx_2 - 0.5) \Big] \geq 0
\end{aligned}$$

Hence, the normalized factor histogram $\bar{h}_f(x)$ of an AC frequency decreases with an increase in the bin index x for uncompressed bitmaps.

For decompressed bitmaps, the block-wise DCT coefficients have been quantized and can be formulated by Eq. (1). According to Eq. (1) and Eq. (2), we have $q \in F(q) \subseteq F(b_n)$ for each b_n, and finally have $\bar{h}_f(x) \equiv 1$ for $x \in F(q)$. In other words, the normalized factor histogram $\bar{h}_f(x)$ of an individual AC frequency from decompressed bitmaps shows maxima at the bin indices $F(q)$. As q is the maximum element of $F(q)$, we can empirically estimate the quantization step by

$$\hat{q} = \max\Big\{ \mathrm{argmax}_x \bar{h}_f(x) \Big\} \tag{6}$$

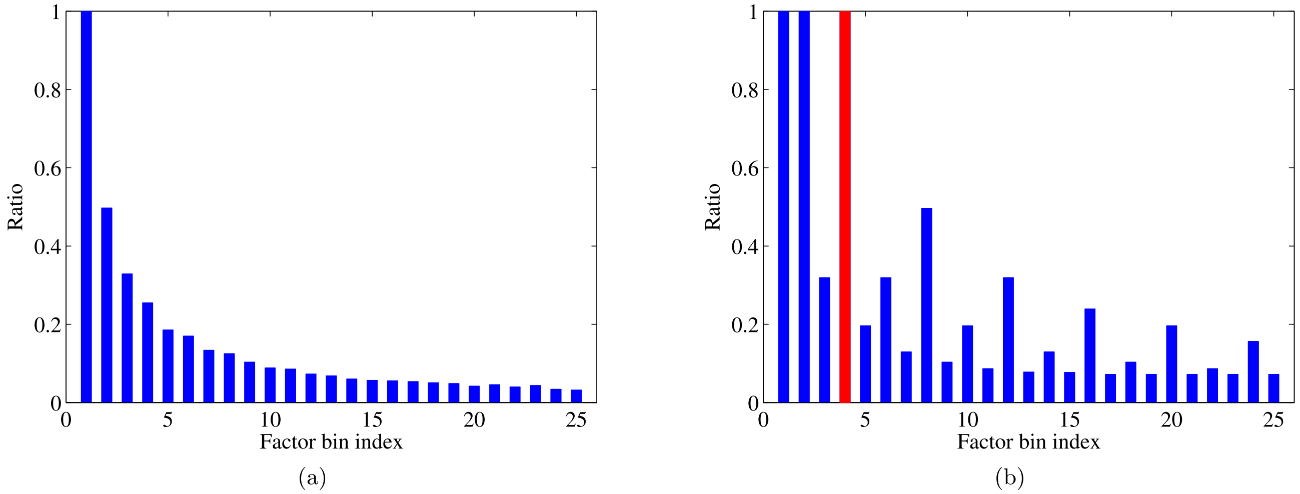
Note that if Eq. (6) is applied to uncompressed bitmaps, the quantization step of each frequency should be estimated to be 1.

Two simulation examples are provided to illustrate the abovementioned characteristics of factor histogram. Given a Gaussian-distributed real sequence $\{a_n\}_{n=1}^N$ with 0 mean and 20 deviation, where $N = 1024$, we round it to an integer sequence $\{b_n^i\}_{n=1}^N$. As a comparison, we also quantize $\{a_n\}_{n=1}^N$ with step 4 and subsequently de-quantize it to generate another integer sequence $\{b_n^q\}_{n=1}^N$. The normalized factor histograms of $\{b_n^i\}_{n=1}^N$ and $\{b_n^q\}_{n=1}^N$ are shown in Fig. 2(a) and Fig. 2(b), respectively. We can see that the normalized factor histograms of $\{b_n^i\}_{n=1}^N$ shows a decreasing trend, while that of $\{b_n^q\}_{n=1}^N$ exhibits maxima at bin indices $F(4) = \{1, 2, 4\}$, and no longer decreases. It is easy to estimate the quantization step by locating the red-labeled bin in Fig. 2(b) according to Eq. (6).

Note that in our previous conference paper [17], we proposed a method for detecting double JPEG compression. Our proposed method in this paper and that in [17] are both based on the distribution of the integer factors of DCT coefficients, but they have significant differences between each other. First, the two methods address different issues. The goal of our previous paper [17] is to detect whether a JPEG image has been doubly compressed or not, and to estimate the quality factor of the primary compression of a doubly compressed image; while this paper focuses on identifying whether a bitmap has been JPEG compressed or not, and estimating the quantization steps of a decompressed bitmap. Second, the factor histograms of the two papers are derived from different models and thus have different expressions. The factor histograms in this and our previous paper [17] are derived from the single and double quantization models of JPEG compression, respectively. Third, these two methods are based on different prior knowledge. The method of [17] is based on the assumption that the series of the quantization tables used for compression is known; while this proposed method is no longer limited to such an assumption. Fourth, this paper provides an analysis on the characteristics of the factor histogram (i.e., its monotonicity and local maximum); while our previous conference paper [17] does not.

3. Proposed method for estimating JPEG compression history

Since each gray-value of a bitmap is usually represented in 8-bit integer, a rounding-and-truncation operation is applied at the end of JPEG decompression to set the output of IDCT (i.e., float gray-values) back to integers of the range [0, 255]. As a result, rounding and truncation errors are inevitably introduced into decompressed bitmaps. It is reported in [10,12] that truncation error can be removed by excluding saturated blocks (i.e., those blocks with gray-value 0 or 255). Similarly, we calculate factor histogram only based on unsaturated blocks (i.e., those blocks without gray-value 0 or 255) in order to avoid the negative effect caused by truncation errors on the estimation of JPEG compression history.



(a)

(b)

Fig. 2. (a) Normalized factor histogram of $\{b_n^i\}_{n=1}^N$. (b) Normalized factor histogram of $\{b_n^q\}_{n=1}^N$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

After excluding all saturated blocks, we adopt Eq. (4) for fast calculation of the factor histogram. In addition to lower computation complexity, another advantage provided by Eq. (4) is that we could perform some preprocessing on DCT coefficient histogram to further enhance the discriminability of factor histogram. Note that $F(0) = \mathbb{N}$, which means that bin 0 of the DCT coefficient histogram contributes to factor histogram in a way that all bins of factor histogram increase the same value. In other words, bin 0 of the coefficient histogram provides no discriminative information. Furthermore, the values of bins 1 and -1 of the coefficient histogram are usually too large, which likely dominate the normalization of the factor histogram. Therefore, we set the values of bins 0, 1 and -1 to be zero.

In summary, the factor histogram of a given bitmap is calculated through the following steps:

1. Collect all 8×8 unsaturated blocks of the given bitmap;
2. Perform block-wise DCT on the collected blocks, and round the resulted float coefficients to their nearest integers, and then calculate the rounded-coefficient histogram;
3. Set the values of bins 0, 1 and -1 of the rounded-coefficient histogram to be 0;
4. Based on the rounded-coefficient histogram obtained in step 3, calculate the factor histogram according to Eq. (4).
5. Normalize the factor histogram according to Eq. (5).

Note that Step 1 was originally proposed in [10] for removing the effect of truncation errors. As done in [10,12], we also implement it as a pre-processing step.

3.1. Identification of decompressed bitmap

As mentioned in Section 2.2, the monotonicity of factor histogram could serve as an indicator of the presence of JPEG compression. We propose a scalar metric to measure the monotonicity of factor histogram, which is given by

$$s \triangleq \max_x \left\{ \bar{h}_f(x) - \bar{h}_f(x-1) \right\}$$

where $\bar{h}_f(x)$ is the normalized factor histogram of an AC frequency. Obviously, s will be negative (or is close to 0) if the factor histogram has a decreasing (or almost decreasing) trend, and will take a relatively large value otherwise. For a given bitmap, instead of calculating $\bar{h}_f(x)$ frequency by frequency, we calculate the normalized factor histogram of the sequence consisting of all AC (rounded) coefficients, denoted by $\bar{h}_f^{ac}(x)$. The monotonicity metric of $\bar{h}_f^{ac}(x)$ is given by

$$s^{ac} \triangleq \max_x \left\{ \bar{h}_f^{ac}(x) - \bar{h}_f^{ac}(x-1) \right\} \tag{7}$$

Since the above assumption on the PDF of DCT coefficients (i.e., $p_{ac}(u) = p_{ac}(-u)$ and $p_{ac}(u_1) \geq p_{ac}(u_2)$ if $|u_1| \leq |u_2|$) does not always hold, s^{ac} may take a small positive value in some cases. Thus, an empirical threshold T^{ac} is applied to make a binary decision: the given bitmap is labeled as "decompressed" if $s^{ac} \geq T^{ac}$, and it is labeled as "uncompressed" otherwise. T^{ac} has a significant impact on the final performance, and it could be trained easily for different image sizes based on a minimum classification error rule.

It should be pointed out that we could also calculate the monotonicity metric s for each AC frequency and then makes a decision based on the resulted 63 metrics. Since the factor histogram $\bar{h}_f^{ac}(x)$ is more stable than the factor histogram $\bar{h}_f(x)$ of each individual frequency, we prefer to use the metric s^{ac} instead of s for identifying decompressed bitmaps.

Fig. 3 illustrates the statistical difference of s^{ac} between uncompressed and decompressed bitmaps. The histogram of s^{ac} of 5000 uncompressed bitmaps with size of 256×256 (the source is referred in Section 4) and that of the corresponding decompressed bitmaps (with quality factor 95) are shown in different colors. As it can be seen, s^{ac} distributes extremely near 0 for uncompressed bitmaps, while it takes relatively large values for decompressed bitmaps. Therefore, the uncompressed and decompressed bitmaps can be easily distinguished from each other based on the metric s^{ac}.

3.2. Estimation of quantization step

Once a decompressed bitmap has been identified, the quantization step of each frequency is desired to be estimated. Due to rounding errors, the factor histogram of each frequency from decompressed bitmaps exhibits a local maximum instead of a global maximum at the bin index which corresponds to the quantization step. To deal with the rounding errors, we slightly modify Eq. (6) to estimate the quantization step by

$$\hat{q} = \max \left\{ x | \bar{h}_f(x) \geq T \right\} \tag{8}$$

where T is an empirical threshold and $0 \leq T \leq 1$. Apparently, Eq. (8) is equivalent to Eq. (6) if $T = 1$. The rationale behind Eq. (8)

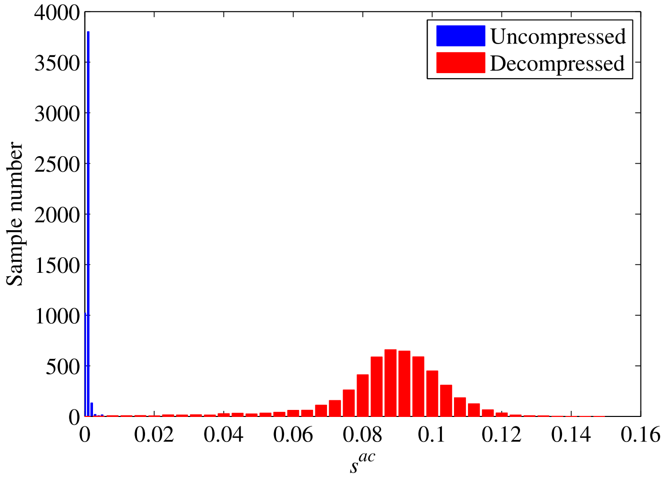


Fig. 3. Histograms of s^{ac} of uncompressed bitmaps and decompressed bitmaps.

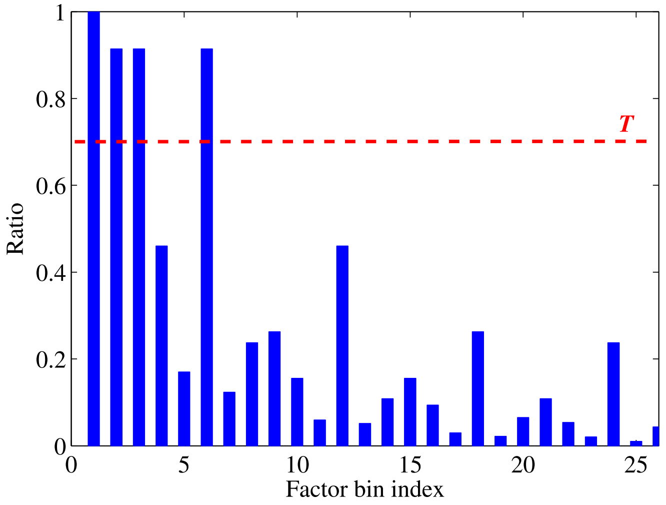


Fig. 4. Normalized factor histogram of DCT coefficients quantized by 6.

is that the factor histogram at the bin index corresponding to the quantization step should still take a relatively large value, despite it may not exactly take the global maximum 1. The optimal T can be trained for each combination of image size and frequency. However, this will make the training task overcomplicated. In our experience, a suboptimal but more universal T could be taken from the range $[0.6, 0.8]$ without significant loss in performance. For simplicity, we set $T = 0.7$ in this paper.

To illustrate Eq. (8), we show the normalized factor histogram of frequency $(1, 0)$ from a decompressed bitmap in Fig. 4. In this example, the coefficients at frequency $(1, 0)$ have been quantized with step 6. Due to the rounding errors occurred during JPEG decompression, the value of bin 6 takes a local instead of a global maximum. We can see that among all the bin indices with value larger than $T = 0.7$, the largest bin index is 6, which is exactly equal to the truth value of the quantization step.

4. Experimental evaluation

For experimental evaluation, 5000 uncompressed images are taken from 5 image datasets, namely, UCID [23], NRCS [24], COREL (from the CorelDraw version 10.0 software CD #3), NJIT (created by Prof. Y.Q. Shi of New Jersey Institute of Technology, USA) and SYSU (taken by Panasonic DMC-FZ30 and Nikon D300 in TIFF format), each of which provides 1000 uncompressed images. All of these uncompressed images are first converted into gray-scale bitmaps. As mentioned in Section 1, tamper detection requires that the methods for estimating JPEG compression history should be able to perform on small image blocks in order to locate the tampered regions. Moreover, most of the test methods can work well for image sizes larger than 256 × 256. Therefore, these gray-scale bitmaps are then centrally cropped into six specific sizes: 256 × 256, 128 × 128, 64 × 64, 32 × 32, 16 × 16, 8 × 8 (pixels), respectively. The JPEG toolbox [25] is applied to perform JPEG compression and decompression. Two sets of experiments are conducted to investigate the performances of our proposed method by comparing with the existing methods.

4.1. Identification of decompressed bitmap

For each image size, these 5000 uncompressed bitmaps are first equally divided into two subsets, one for training a proper threshold T^{ac}, and the other for testing. In the training stage, each bitmap in the training subset is compressed with a quality factor (QF) randomly selected from $\{50, 51, \cdots, 99\}$ with equal probability, and then decompressed back into a bitmap. Since the QF 100 corresponds to an all-one quantization step matrix, it is essentially indistinguishable and thus is not evaluated in our experiment. The optimal threshold is searched within the range $[0, 1]$ with a step size of 0.0001, and the threshold corresponding to the highest identification accuracy (i.e., the percentage of bitmaps being correctly labeled) is selected as the optimal T^{ac}.

Our proposed method is first evaluated for several representative QFs. The curves of identification accuracy are demonstrated in Fig. 5. It is shown that the proposed method works quite well for QF ≤ 95, where the identification accuracy is higher than 90% for image size as small as 16 × 16. Even when the QF increases to 98, the identification accuracy is still up to 90% for image sizes larger than 64 × 64. However, the identification accuracy falls below 90% for all investigated image sizes, when QF $= 99$. This is also the hardest case that the existing methods can detect, because 41 out of 63 AC quantization steps for QF $= 99$ are equal to 1, which means that most of the AC frequencies do not contribute any discriminative information. Moreover, we test the case of QF $= 10$ in order to evaluate the performance of the proposed method for extremely low QF. Note that the threshold of the proposed method is trained without using bitmaps compressed by QF $= 10$ in this experiment. We also conduct another experiment to train a threshold by including the samples of QF $= 10$ and then use the threshold for identification. In these two experiments, the identification accuracies are similar, which means that the threshold is insensitive to the training samples. Fig. 5 shows that the identification accuracies of QF $= 10$ are lower than those of QF $= 50$, 65, 80 for small image sizes. The reason could be that there are too few non-zero DCT coefficients in small and heavily compressed bitmaps. In brief, the proposed feature s^{ac} performs well for most combinations of quality factor and image size, which indicates the factor histogram is feasible and effective for identifying decompressed bitmap.

We then compare the proposed method with the state-of-the-art methods, namely, Fan et al.'s [10], Fu et al.'s [11] and Luo et al.'s [12]. The average identification accuracies over all of the QFs $\{50, 51, \cdots, 99\}$ are listed in Table 1. The performance of Fan et al.'s method is limited. In their algorithm, they tried to measure the blocking strength in pixel domain, and this could be easily affected by the inherent content edge of the test bitmap. Luo et al.'s feature is especially applicable for small image sizes, such as 8 × 8, however, its performance is not that remarkable for larger image sizes. Compared to these existing methods, our proposed method is superior to Fan et al.'s and Fu et al.'s methods for all investigated image sizes, while it achieves higher identification accuracy than Luo et al.'s method for image size 32 × 32 or larger. This out-

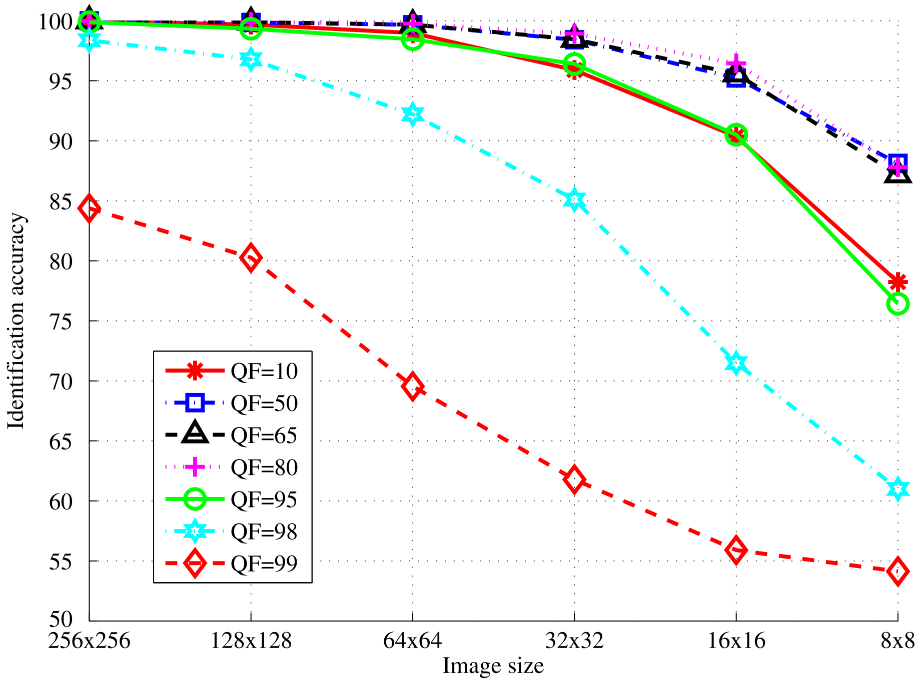


Fig. 5. Identification accuracy (%) curves of the proposed method in identifying decompressed bitmap.

Table 1
Identification accuracies (%) of the proposed and the compared methods.

Methods	Image size (pixels)					
	256^2	128^2	64^2	32^2	16^2	8^2
Fan et al.'s [10]	90.30	80.24	64.20	54.72	50.98	N/A
Fu et al.'s [11]	98.44	98.84	98.06	94.06	87.68	79.64
Luo et al.'s [12]	98.02	97.80	97.58	97.16	96.74	94.04
Proposed	99.94	99.74	99.14	97.60	93.16	83.46

come indicates that our proposed method could perform best for identifying decompressed bitmaps with size greater than 32 × 32.

4.2. Estimation of quantization step

In the second set of experiments, each of 5000 uncompressed bitmap is JPEG compressed with a QF randomly selected from $\{50, 51, \cdots, 99\}$ with equal probability, and then decompressed back into a bitmap. The quantization steps of these decompressed bitmaps are estimated by our proposed and the existing methods, and the results are demonstrated below. Note that a single 8 × 8 block is insufficient for estimating the quantization steps. The case for 8 × 8 blocks is therefore not included in the results.

The estimation accuracy (i.e., the percentage of the estimated quantization step being exactly equal to the true quantization step) of the proposed method for each individual frequency is first evaluated. In Fig. 6, for image size 256 × 256, 30 out of 64 frequencies achieve estimation accuracy of up to 95%, and another 12 frequencies achieve estimation accuracy of up to 85%. These outcomes indicate that our proposed method is applicable for a wide range of frequencies. When the frequency increases, the estimation accuracy decreases to some extent. The reason is that the high-frequency coefficients are quantized by larger steps. As a result, more coefficients are quantized to be zeros, and thus less information is available for estimation.

We then compare our proposed method with other state-of-the-art methods, namely, Neelamani et al.'s [13], Ye et al.'s [8], Luo et al.'s [12] and Lin et al.'s [9]. For a fair and meaningful comparison, we evaluated the results over the first 20 AC frequencies in zigzag order, as Luo et al.'s method fails for direct-current frequency and all of the compared methods are considerably unreliable for higher AC frequencies. The average estimation accuracy over the QFs $\{50, 51, \cdots, 99\}$ and the investigated 20 AC frequencies are listed in Table 2. Our proposed method achieves average estimation accuracies from 96.58% to 99.24% for image sizes from 64 × 64 to 256 × 256, which is 4%–10% more accurate than the best one of the other compared methods. For image size 32 × 32, as the performance of other methods falls to an unacceptable level, our proposed method could still maintain an accuracy of 88% on

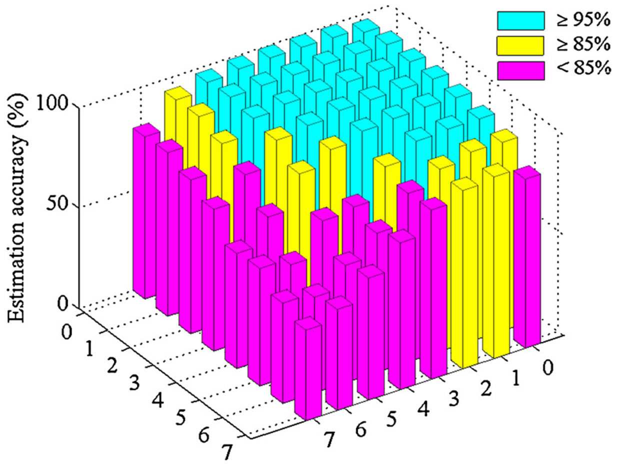


Fig. 6. Estimation accuracies (%) of the proposed method for individual frequencies (image size: 256 × 256).

Table 2
Average estimation accuracies (%) of the proposed and the compared methods over the first 20 AC frequencies in zigzag order.

Methods	Image size (pixels)				
	256^2	128^2	64^2	32^2	16^2
Neelamani et al.'s [13]	90.17	89.22	86.83	82.12	62.74
Ye et al.'s [8]	82.46	79.51	61.79	38.32	17.89
Luo et al.'s [12]	92.85	85.51	74.11	59.48	43.71
Lin et al.'s [9]	95.04	91.91	84.48	57.69	21.14
Proposed	99.24	98.35	96.58	88.16	60.15

Table 3
Average elapsed time (millisecond per bitmap) for identifying decompressed bitmap.

Methods	Image size (pixels)					
	256^2	128^2	64^2	32^2	16^2	8^2
Fan et al.'s [10]	1.07	0.35	0.25	0.23	0.19	0.02
Fu et al.'s [11]	16.55	3.73	1.53	0.98	0.77	0.77
Luo et al.'s [12]	10.96	2.46	0.92	0.69	0.60	0.56
Proposed	14.09	3.39	1.46	1.06	0.97	0.84

Table 4
Average elapsed time (millisecond per bitmap) for estimating quantization step.

Methods	Image size (pixels)				
	256^2	128^2	64^2	32^2	16^2
Neelamani et al.'s [13]	206.95	196.42	194.32	189.85	179.82
Ye et al.'s [8]	22.39	14.71	12.69	11.72	9.45
Luo et al.'s [12]	18.55	10.89	8.90	8.09	6.34
Lin et al.'s [9]	25.13	14.74	12.62	11.56	9.40
Proposed	28.10	16.45	11.95	9.25	6.95

average. These advantages could be attributed to the fact that factor histogram well reveals the trace of quantization. However, our proposed method shows its limitation for image size 16 × 16. The reason is that a single 16 × 16 block only provides at most 4 non-zero coefficients for each frequency, so a rounding error would probably lead to a failure of estimation.

In addition, we also briefly evaluate the runtime of our proposed method. All algorithms are implemented in Matlab and run on a PC with a 2.8 GHz CPU and 2 GB RAM. In the aspect of decompressed bitmap identification [10–12], the elapsed time per bitmap is listed in Table 3. Fan et al.'s method [10] is the fastest of all test methods as it is based on measuring blocking artifacts directly in pixel domain albeit at the cost of low identification accuracy. The runtime of our proposed method is of the same order as those of Fu et al.'s [11] and Luo et al.'s [12], which is only 10-millisecond scale for image size 256 × 256. In the aspect of quantization step estimation [8,9,12,13], the elapsed time per bitmap is shown in Table 4. Neelamani et al.'s method [13] is relatively time-consuming because it needs to calculate the values of a complicated likelihood function for all candidate steps. In contrast, our proposed and the other three methods [8,9,12] only require about one tenth of elapsed time of Neelamani et al.'s method. Such competitive time efficiency of our proposed method is due to the simplicity of the estimation rules (i.e., Eq. (8)) and the accelerated calculation of the factor histogram (i.e., Eq. (4)).

5. Conclusions

The estimation of JPEG compression history for bitmaps has been widely studied in the past decade due to its significant role in the applications of de-blocking, smart recompression, image forensics and steganalysis. In this paper, we have proposed a novel factor histogram based method for estimating JPEG compression history of bitmaps. The contributions can be summarized as follows.

First, by formulating the quantization and de-quantization in the generation of JPEG decompressed bitmap, we have proposed a novel statistic named factor histogram and analyzed its characteristics. Factor histogram is statistically decreasing for an uncompressed bitmap, whereas it exhibits a relatively large bin corresponding to the quantization step for a decompressed bitmap. These characteristics can serve as a robust indicator of the presence of JPEG compression and the clues of quantization step estimation.

Second, based on factor histogram, we have proposed to identify decompressed bitmaps by measuring the monotonicity of factor histogram, and to estimate the quantization step of each frequency by selecting a histogram bin with the maximum index from those bins larger than a threshold. Extensive experiments have been conducted to validate the feasibility and effectiveness of our proposed method, and it outperforms the existing methods for most investigated image sizes.

Our proposed method is originally designed for gray-scale bitmaps in this paper, but it can also be extended to color bitmaps. Estimating the JPEG compression history of color bitmaps has three tasks, including determining the color space, identifying the subsampling rate, and estimating the quantization steps. Note that our proposed method could only perform the last but most important task, i.e., estimating the quantization steps. Fortunately, the paper [13] proposed a framework to perform all of the three tasks. A possible way to extend our proposed method to color bitmaps is to incorporate it into the framework of [13]. In addition, the efficiency of our proposed method is attributed to its capability of detecting the quantization artifacts caused by JPEG compression. JPEG2000, like JPEG, also causes quantization artifacts on its transformed coefficients. Therefore, our proposed method with some modification (e.g., the factor histogram is calculated based on wavelet coefficients instead of DCT coefficients) could be applied to bitmaps compressed by JPEG2000.

Acknowledgments

The authors are grateful to the anonymous referees for their valuable comments. This work has been supported by the National Science and Technology Pillar Program (2012BAK16B06), NSFC (U1135001, 61003297, 61332012, 61202415), the NSF of Guangdong Province (S2013010011806), the NSF of Tanjin (15JCQNJC00-700), the Shenzhen Peacock Program (KQCX20120816160011790), and the Knowledge Innovation Program of Shenzhen (JCYJ201304-01170306848).

References

[1] A. Piva, An overview on image forensics, ISRN Signal Process. 2013 (2013), Article ID 496701.

[2] G.A. Triantafyllidis, D. Tzovaras, M.G. Strintzis, Blocking artifact detection and reduction in compressed data, IEEE Trans. Circuits Syst. Video Technol. 12 (2002) 877–890.

[3] S. Liu, A.C. Bovik, Efficient DCT-domain blind measurement and reduction of blocking artifacts, IEEE Trans. Circuits Syst. Video Technol. 12 (2002) 1139–1149.

[4] Z. Fan, R. Eschbach, JPEG decompression with reduced artifacts, in: Proc. Symp. Electronic Imaging: Image and Video Compression, 1994, pp. 50–55.

[5] H.H. Bauschke, C.H. Hamilton, M.S. Macklem, J.S. McMichael, N.R. Swart, Recompression of JPEG images by requantization, IEEE Trans. Image Process. 12 (2003) 843–849.

[6] J. Fridrich, M. Goljan, R. Du, Steganalysis based on JPEG compatibility, in: Proc. SPIE Multimedia Systems and Applications IV, 2001, pp. 275–280.

[7] J. Kodovskỳ, J. Fridrich, JPEG-compatibility steganalysis using block-histogram of recompression artifacts, in: Proc. Int. Workshop Information Hiding, 2013, pp. 78–93.

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

[9] G.-S. Lin, M.-K. Chang, Y.-L. Chen, A passive-blind forgery detection scheme based on content-adaptive quantization table estimation, IEEE Trans. Circuits Syst. Video Technol. 21 (2011) 421–434.

[10] Z. Fan, R.L. de Queiroz, Identification of bitmap compression history: JPEG detection and quantizer estimation, IEEE Trans. Image Process. 12 (2003) 230–235.

[11] D. Fu, Y.Q. Shi, W. Su, et al., A generalized Benford's law for JPEG coefficients and its applications in image forensics, in: Proc. SPIE Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents, 2007, p. 65051L.

[12] W. Luo, J. Huang, G. Qiu, JPEG error analysis and its applications to digital image forensics, IEEE Trans. Inf. Forensics Secur. 5 (2010) 480–491.

[13] R. Neelamani, R. De Queiroz, Z. Fan, S. Dash, R.G. Baraniuk, JPEG compression history estimation for color images, IEEE Trans. Image Process. 15 (2006) 1365–1378.

[14] J. Lukáš, J. Fridrich, Estimation of primary quantization matrix in double compressed JPEG images, in: Proc. Digital Forensic Research Workshop, 2003, pp. 5–8.

[15] A.C. Popescu, H. Farid, Statistical tools for digital forensics, in: Proc. Int. Workshop Information Hiding, 2005, pp. 128–147.

[16] T. Pevny, J. Fridrich, Detection of double-compression in JPEG images for applications in steganography, IEEE Trans. Inf. Forensics Secur. 3 (2008) 247–258.

[17] J. Yang, G. Zhu, J. Huang, Detecting doubly compressed JPEG images by factor histogram, in: Proc. Asia-Pacific Signal and Information Processing Association, Annual Summit and Conference (APSIPA ASC), 2011.

[18] T. Bianchi, A. Piva, Detection of nonaligned double JPEG compression based on integer periodicity maps, IEEE Trans. Inf. Forensics Secur. 7 (2012) 842–848.

[19] T. Bianchi, A. Piva, F. Perez-Gonzalez, Near optimal detection of quantized signals and application to JPEG forensics, in: Proc. Int. Workshop on Information Forensics and Security (WIFS), 2013, pp. 168–173.

[20] F. Galvan, G. Puglisi, A.R. Bruna, S. Battiato, First quantization coefficient extraction from double compressed JPEG images, in: Proc. Int. Conf. Image Analysis and Processing (ICIAP), 2013, pp. 783–792.

[21] G. Puglisi, A.R. Bruna, F. Galvan, S. Battiato, First jpeg quantization matrix estimation based on histogram analysis, in: Proc. Int. Conf. Image Processing (ICIP), 2013, pp. 4502–4506.

[22] E.Y. Lam, J.W. Goodman, A mathematical analysis of the DCT coefficient distributions for images, IEEE Trans. Image Process. 9 (2000) 1661–1666.

[23] G. Schaefer, M. Stich, UCID: an uncompressed color image database, in: Proc. SPIE Storage and Retrieval Methods and Applications for Multimedia, 2003, pp. 472–480.

[24] U.S. Department of Agriculture, NRCS photo gallery [online]. Available at http://photo-gallery.nrcs.usda.gov, 2005.

[25] P. Sallee, Matlab JPEG toolbox [online]. Available at http://www.philsallee.com/jpegtbx/index.html, 2004.

Jianquan Yang received the B.S. and M.S. degrees from Sun Yat-sen University, Guangzhou, China, in 2008 and 2010, respectively. He is currently an Engineer with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. His research interests mainly include multimedia security and image processing.

Guopu Zhu received the B.S. degree from Jilin University, China, in 2002, and the M.S. and Ph.D. degrees from the Harbin Institute of Technology, China, in 2004 and 2007, respectively. He was a Post-Doctoral Fellow with Sun Yat-sen University, China. He was a Research Associate and Senior Research Associate with the City University of Hong Kong, Hong Kong. Since 2010, he has been an Associate Professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences (CAS). His main research areas are multimedia security and image processing. He has authored about 20 international journal papers in these areas. He is a senior member of IEEE, and also is a member of the Youth Innovation Promotion Association, CAS.

Jiwu Huang 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 served as an Associate Editor of the IEEE Transactions on Information Forensics and Security and a member of Information Forensics and Security Technical Committee, IEEE Signal Processing Society. He was General Co-chair of the IEEE International Workshop on Information Forensics and Security 2013.

Xi Zhao received the B.Sc. (Hons) and M.Sc. degrees in Information Systems from the University of Greenwich, UK in 2006 and University of Surrey, UK in 2007, respectively. In 2011, he received the Ph.D. degree in digital watermarking and image authentication from the University of Surrey, UK, and then joined the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China. In 2014, he joined the School of Computer Sciences and Information Engineering, Tianjin University of Science and Technology, China. He is currently an Associate Professor and his research interests include information hiding, digital watermarking, image authentication, image forensics and biometric security.