Digital Image Source Coder Forensics Via Intrinsic Fingerprints

W. Sabrina Lin, *Member, IEEE*, Steven K. Tjoa, H. Vicky Zhao, *Member, IEEE*, and K. J. Ray Liu, *Fellow, IEEE*

***Abstract*—Recent development in multimedia processing and network technologies has facilitated the distribution and sharing of multimedia through networks, and increased the security demands of multimedia contents. Traditional image content protection schemes use extrinsic approaches, such as watermarking or fingerprinting. However, under many circumstances, extrinsic content protection is not possible. Therefore, there is great interest in developing forensic tools via intrinsic fingerprints to solve these problems. Source coding is a common step of natural image acquisition, so in this paper, we focus on the fundamental research on digital image source coder forensics via intrinsic fingerprints. First, we investigate the unique intrinsic fingerprint of many popular image source encoders, including transform-based coding (both discrete cosine transform and discrete wavelet transform based), subband coding, differential image coding, and also block processing as the traces of evidence. Based on the intrinsic fingerprint of image source encoders, we construct an image source coding forensic detector that identifies which source encoder is applied, what the coding parameters are along with confidence measures of the result. Our simulation results show that the proposed system provides trustworthy performance: for most test cases, the probability of detecting the correct source encoder is over 90%.**

***Index Terms*—Image source coding, intrinsic fingerprint, multimedia forensics.**

I. Introduction

WITHIN the past decades, the explosive combination of multimedia signal processing, communications, and networking technologies has facilitated the sharing of digital multimedia data and enabled pervasive digital media distribution. Digital images, in particular, have been widely used in news reporting, insurance claim investigation, criminal investigation, and many other applications. However, the digital nature of information also allows individuals to access, duplicate, or manipulate information beyond the terms and the conditions agreed upon, for example, with the assistance of popular image editing software such as Adobe Photoshop. Therefore, the traditional faith that a photograph tells the truth is diminished for digital images, and it raises critical issues about the legitimacy and authenticity of the image.

Manuscript received March 03, 2009; revised May 27, 2009. First published June 05, 2009; current version published August 14, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Nasir Memon.

W. S. Lin, S. K. Tjoa, and K. J. R. Liu are with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA (e-mail: wylin@umd.edu; kiemyang@umd.edu; kjrliu@umd.edu).

H. V. Zhao is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, T6G 2V4 Canada, (e-mail: vzhao@ece.ualberta.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2009.2024715

Image forensics offers the ability to verify the credibility and validate the origin of digital images. Conventional forensic technologies use proactive and additive means to protect multimedia content by hiding additional information in the original signal. For example, the idea of the trustworthy camera was proposed in [1] to make the trustworthiness of digital images accountable, where a digital watermark was embedded into the image at the instance of its acquisition. Any later tampering of the image can be detected based on the changes on the digital watermark. Similarly, in traitor-tracing digital fingerprinting, user identification information is embedded in each distributed copy to identify the corresponding user and trace the source of the illicit copies [2]. However, they require that all camera manufacturers agree upon a common standard, and for some real applications, it may be too expensive and impractical to implement such *extrinsic* protection mechanisms.

Often it is not possible to enforce content protection through any extrinsic means. However, for each copy of multimedia data, its acquisition, processing, and transmission process constitutes a unique data path. To ensure that multimedia data are processed by the appropriate entities for intended purposes only, its data path must be validated by identifying each of its steps: acquisition, source coding, channel coding, transmission, and other possible processing at the user's side. Each operation leaves its unique artifact in the image. Such intrinsic "fingerprints" are naturally and inherently generated throughout the chain of content acquisition and processing, and they provide evidence to help identify the origin and detect the alterations of multimedia content. Thus in the scenarios where extrinsic content protection techniques are not applicable, *image forensics via intrinsic fingerprints* offers technologies to detect alterations and identify the source of the image without any proactive protection mechanisms. Fig. 1 illustrates the difference between the methodologies that rely on extrinsic operations to protect multimedia content versus those that employ intrinsic fingerprint analysis. With extrinsic protection of multimedia, an additive signal is embedded into the image before distribution and is available to the forensic detector, while with intrinsic-fingerprint forensics, the only input to the forensic detector is the received image in a raw format.

Image forensics via intrinsic fingerprints can be applied to identify many types of image processing. For instance, it is often useful to determine the source of image acquisition: a digital camera or a postediting software. Since most image acquisition devices reduce the data size by applying lossy encoders to the images, there are prior arts in the literature which identify the camera model based on the JPEG quantization table [3]. Forensics on other steps in image compression such as block size estimation [4]–[7] via the intrinsic fingerprints have also been studied in the literature. In this paper, we build a framework to integrate the image source encoding forensics that provides a

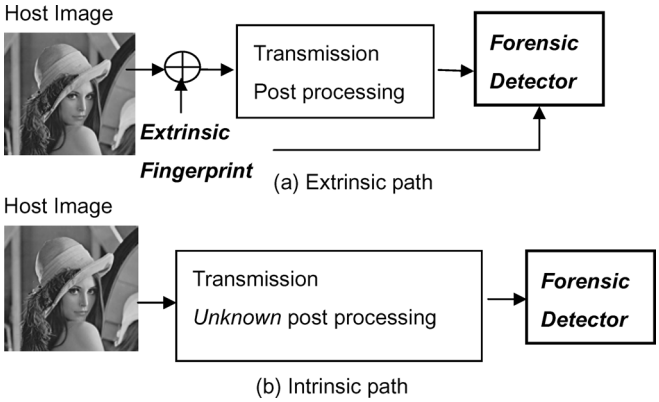


Fig. 1. Multimedia security using extrinsic means versus intrinsic information.

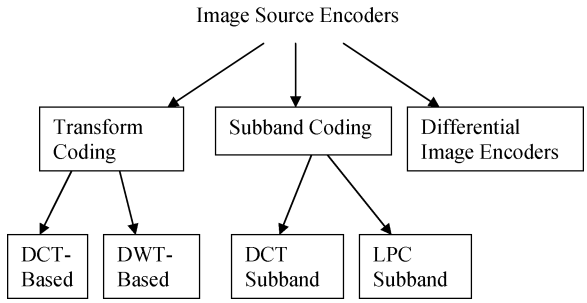


Fig. 2. Tree-structure of the image coding forensics system.

general methodology and fundamental research of image source coding identification.

There exists some literature devoted to identify the unique features associated with each form of image processing, for example, resampling [8], inconsistent noise patterns [9], copy and paste [10], double compression [11], [12], etc. Inconsistency in higher order statistics, such as the bispectrum and bicoherence [13], were used to identify contrast changes such as gamma correction [14] and other nonlinear operations on images [15]. Inconsistency in lighting conditions [16], geometry invariants, and consistency of camera characteristics [17] were also proposed to detect image alteration. The work in [18] used wavelet-based features to detect image tampering. Physics-motivated features were introduced to distinguish photographic images and computer graphics [19]. To identify the source of an image, pixel defects [20] and image sensor noise [21] were used to uniquely identify the source camera. Features such as RGB-pair correlation and neighbor center of mass [22] were extracted as features and a support vector machine were used to train classifiers to identify the camera. In [23] and [24], color filter array and color interpolation were used to classify the camera brands.

This paper investigates the identification of source coding algorithms which is often an unavoidable step in image acquisition and transmission. Each compression scheme leads to different types of distortions. These distortions become an intrinsic part of the compressed image, acting like a fingerprint. By differentiating amongst these distortion types, we can identify the scheme and estimate the parameters used to compress a digital image.

Source coder identification has many applications in multimedia security, coding, and communication, particularly when we lack access to the original signal or the device. As mentioned earlier, we can verify the *datapath integrity* of multimedia data. For example, consider two datapaths A and B. Image X_A travels over datapath A and image X_B travels over datapath B. Each datapath consists of different digital cameras used for image acquisition, different source coders, and different transmission channels. We can certify the datapath taken by possibly examining evidence of a particular color filter array used by the camera, the intrinsic fingerprints produced by the source coder, and traces of error concealment due to channel errors or network failure.

Also, *digital image integrity* is of paramount importance in many forensic scenarios. The Scientific Working Group on Imaging Technologies—part of the International Association for Identification, an organization devoted to forensic science—cites potential legal ramifications regarding the use of digital image processing in forensic contexts [25]. For example, compression is often an unavoidable step in the image acquisition process. The compression history of an image may become relevant in judicial proceedings, since one could argue that compression artifacts had obscured relevant information.

Unfortunately, the compression algorithm and settings may not be immediately obvious, especially if performed automatically as a result of the acquisition device (e.g., compression in digital cameras). In this case, when the compressor is unknown and possibly proprietary, is there any way to determine the compression algorithm? This information is critical in subsequent quantitative image analysis, where the use of image compression can degrade the accuracy of object measurements. Such inaccuracies could lead to an incorrect diagnosis from a medical image, or an incorrect statement of guilt regarding a subject involved in a crime as viewed by a surveillance camera. Through nonintrusive forensic analysis, we can identify the nature of the processing module in the absence of the original image, thereby offering some measure of confidence regarding subsequent image analysis.

The primary objective of this paper is to introduce a forensic methodology that uses intrinsic fingerprints to detect traces and identify the history of source coding operations applied to digital images. Section II formulates the problem and introduces the system model. In Section III, we address the identification of preprocessing applied to the image before compression. In Section IV, for each source encoder studied in this paper, we analyze its unique intrinsic fingerprint and propose a similarity measure to quantify the likelihood that an image is compressed using this source coding scheme. In Section V, we propose an image source coding forensic detector that uses intrinsic fingerprints to identify the source coding scheme used to compress an image, estimate the coding parameters, and provide confidence measurement of the detection results. Section VI shows simulation results. Conclusions are drawn in Section VII.

II. Problem Formulation and System Model

Given a received and decoded image which has been source-encoded once, we analyze the image in order to answer which compression scheme was used to compress the image, if any preprocessing such as blocking was performed before compression, what the parameters of the coding scheme are, as well as how confident we are of the detection and estimation results.

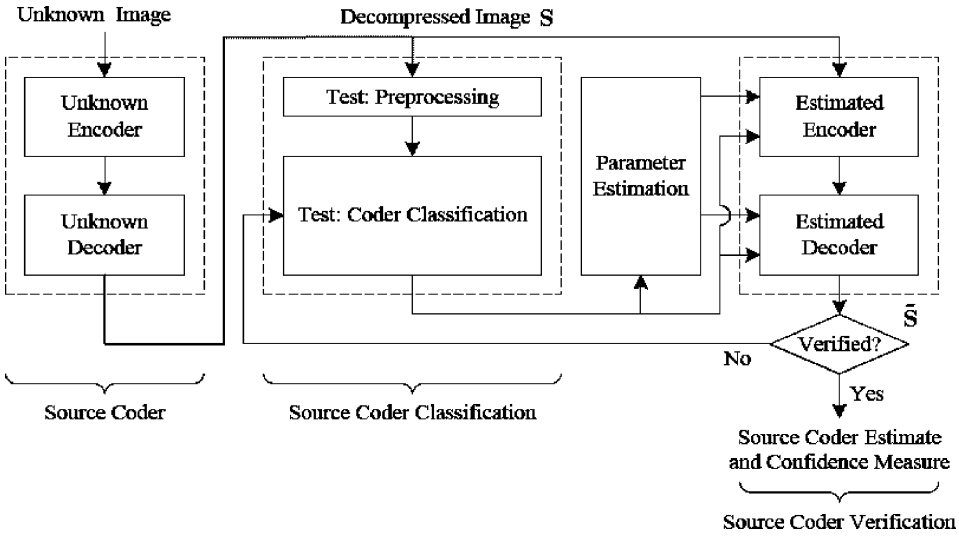


Fig. 3. Overview of the proposed system model for source coder identification.

To answer the above questions, the first step is to categorize the major source encoders. Today, discrete cosine transform (DCT)-based encoders [26] such as JPEG are the most widely used image encoding schemes. Discrete wavelet transform (DWT)-based encoders like JPEG2000 and SPIHT are also widely used in modern image source coding, and they can be viewed as special realizations of subband encoders which share the same intrinsic fingerprints. Therefore, to make our system more generalized, it is important to investigate the common intrinsic fingerprint of subband coding [27]. Differential image encoders remove the redundancy among pixels by spatial filtering. This technology is still in use today as seen in the intraprediction mode in H.264 [28], video interframe prediction, and lossless JPEG. Therefore, to provide a fundamental technology, our forensic system includes these three source coding schemes, as shown in Fig. 2.

Fig. 3 illustrates our proposed system model. All of the test images used throughout this work are digital grayscale images with eight bits per pixel. Given a test image $\mathbf{S}$, we develop an iterative source coder identification and verification system. The first step of the identification process is to identify any preprocessing performed upon the image before compression. Then, for each candidate source coding scheme in Fig. 2, we look for its trace in the test image and calculate its similarity measure (that is, the likelihood that the test image was compressed using this source coding scheme). We then select the one that gives the highest similarity measure and estimate the coding parameters.

The next step is to verify our estimation result. If we select the correct source coding scheme and accurately estimate all of the coding parameters during the classification stage, then if we compress and decompress the input test image $\mathbf{S}$ using the selected source coding schemes with the estimated parameters, then the output $\bar{\mathbf{S}}$ will be identical to the test image $\mathbf{S}$ except for rounding errors. When the difference between $\mathbf{S}$ and $\bar{\mathbf{S}}$ is smaller than a threshold, we stop the search process and output the confidence measure of the system. Otherwise, we take this source coding scheme out of our search space, select another candidate in Fig. 2 with the second highest similarity measure, and repeat the classification and the parameter estimation processes again until we find one solution that satisfies the stopping criteria. If we search over the entire image coding scheme set and none of them can pass the verification stage, the system outputs the error message "No source coding scheme detected."

III. Image Block Processing Identification

As shown in Fig. 3, to begin the forensic process, we must first address any form of preprocessing performed upon the image. Detection of preprocessing is an obvious and crucial first step, because any inaccuracies at this stage could possibly invalidate all subsequent tests. The most common form of image preprocessing is block processing, which we focus on here.

Existing work in block processing measurement is not tailored to answer this question due to strong assumptions placed upon the input data. Works such as Minami and Zakhor [4], Tan and Ghanbari [5], Liu and Bovik [6], and Gao *et al.* [7] assume *a priori* that the image data is compressed through an established scheme such as JPEG or MPEG with some known block size. In a forensic scenario, we have no idea of the block size. Given this problem, we need a scheme that does not rely on such strong assumptions regarding the block size.

A. Intrinsic Fingerprint Analysis

Blocking artifacts may appear as a result of coarse quantization of individual blocks either in the spatial domain or the transform domain. Since quantization is performed on each block separately, a boundary will appear between the blocks as an abrupt change in the luminance value. For an original, unquantized image, this artifact is unlikely to appear because natural images have smooth variations, on the whole. Considering the luminance discontinuities present across block boundaries, the gradient magnitude image should reveal the presence of block artifacts.

Therefore, after computing the gradient magnitude of each row and column in a block-processed image, we expect to find peaks at every block boundary position. We can further reveal these peaks by averaging the gradient magnitudes of all rows (or columns) together. Let X be the input image of size $M \times N$, and $X(i,j)$ be the luminance value of pixel (i,j), where $i \in \{0,\ldots,M-1\}$ and $j \in \{0,\ldots,N-1\}$. First, obtain the gradient image along a desired direction. For example, we operate

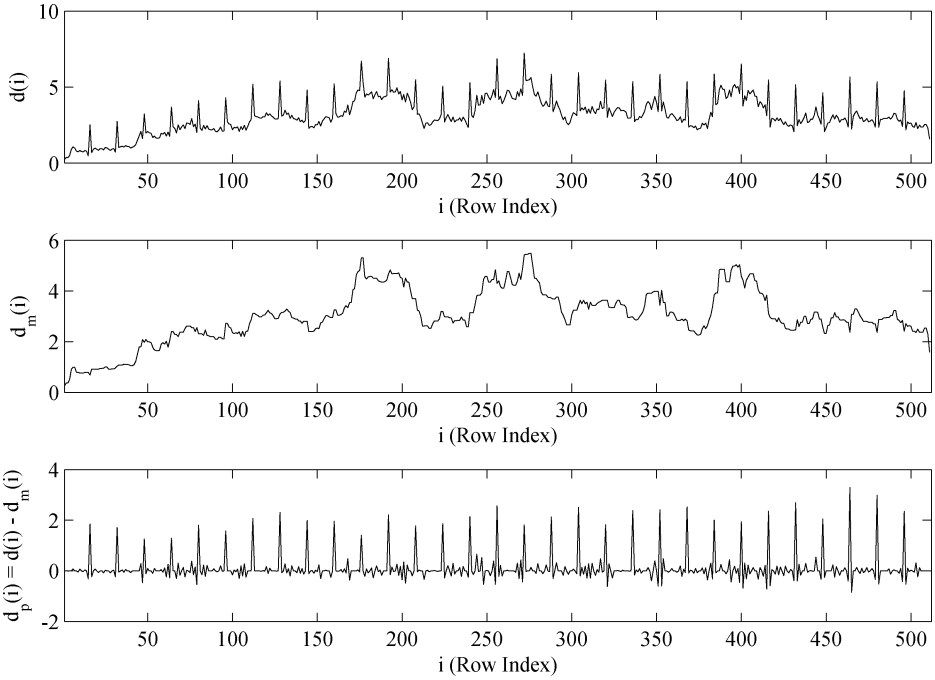


Fig. 4. Signals $\mathbf{d}$, $\mathbf{d}_m$, and $\mathbf{d}_p$ for JPEG-compressed Lena with a block size of 16.

along the vertical direction to obtain $D(i,j) = X(i,j) - X(i-1,j), \quad i \in \{1,\ldots,M-1\}$.

Then obtain the average of the gradient magnitude by averaging along the orthogonal direction. For example, we now average along the horizontal direction, where

$$d(i) = \frac{1}{N}\sum_{j=0}^{N-1} |D(i,j)|, \quad i \in \{1,\ldots,M-1\}. \tag{1}$$

Let B be the block size, if it exists. If block processing is present, the one-dimensional signal $\mathbf{d}$ will have peaks at multiples of the block size (i.e., at $i = kB$, for $k \in \mathbb{Z}$). To extract these peaks, we use median filtering as follows. Let $\mathbf{d}_m$ be the median-filtered version of $\mathbf{d}$: $d_m(i) = \text{median}\,\{d(i-1), d(i), d(i+1)\}$. If we subtract $\mathbf{d}_m$ from $\mathbf{d}$ itself, we will obtain the peaks in $\mathbf{d}$ which we call $\mathbf{d}_p$, the intrinsic block artifact fingerprint

$$d_p(i) = d(i) - d_m(i). \tag{2}$$

Fig. 4 shows the signals $\mathbf{d}$, $\mathbf{d}_m$, and $\mathbf{d}_p$ for the 512-by-512 test image *Lena* which has been JPEG-compressed using a block size of 16 and standard JPEG compression table. We expect $\mathbf{d}_p$ to resemble an impulse train, where the magnitude of the impulses is determined by the strength of the block artifacts, and the period of the impulses is determined by the block size.

One problem we face is the presence of spurious peaks in the signal $\mathbf{d}_p$ as a result of edges from objects in the image. Note that an edge will have the same gradient direction along its entire length. However, the gradient direction of block artifacts will oscillate. Therefore, we perform the following test. Let the signal $c(i)$ be a sum of the gradients

$$c(i) = \frac{1}{N}\sum_{j=0}^{N-1} D(i,j), \quad i \in \{1,\ldots,M-1\}. \tag{3}$$

A peak in this signal will indicate the presence of an edge. Therefore, we find $\mathbf{c}_p = \mathbf{c} - \mathbf{c}_m$, where $\mathbf{c}_m$ is a median-filtered version of $\mathbf{c}$. Then we set $d_p(i) = 0$ for all i, where $c_p(i) > \tau$, for some suitable threshold τ. For the images used in our data set, the value $\tau = 5.0$ maximizes the number of correctly estimated block sizes among all images tested. Images which use more or less than eight bits per pixel may require a different threshold; this value can be chosen by the system designer based upon the nature of the images processed.

B. Block Size Estimation

Fig. 5 shows the block diagram for our detection scheme. The periodicity of $\mathbf{d}_p$ allows us to use a maximum-likelihood estimation scheme used in pitch detection [29] to determine the period of the signal $\mathbf{d}_p$. Suppose that $\mathbf{d}_p$ consists of a known periodic signal $\mathbf{s}$ plus zero-mean i.i.d. Gaussian noise

$$d_p(i) = s(i) + n(i), \quad i \in \{1,\ldots,M-1\}. \tag{4}$$

Let us express the signal $\mathbf{s}$ as a periodic repetition of a signal $\mathbf{q}$ with period B

$$s(i) = q(i \bmod B). \tag{5}$$

To obtain the maximum-likelihood estimate, we maximize the conditional probability density function $p(\mathbf{d}_p|\mathbf{s},\sigma^2,B)$ with respect to the signal parameter $\mathbf{s}$, the noise variance σ^2, and the period B. It can be shown that the estimated period $\hat{B}$ that maximizes $p(\mathbf{d}_p|\mathbf{s},\sigma^2,B)$ is achieved by minimizing the estimated noise variance $\hat{\sigma}^2(B)$ as a function of B

$$\hat{B} = \text{argmin}_B \hat{\sigma}^2(B). \tag{6}$$

This is our estimate for the block size along one dimension (e.g., the vertical or horizontal dimension). We repeat the process for the other dimension to obtain the estimate for the block size in both dimensions.

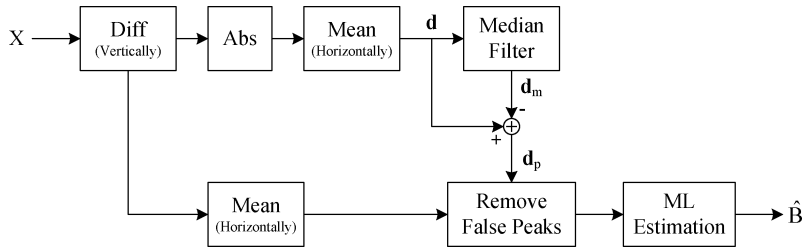


Fig. 5. Block diagram for the block artifact fingerprint detection and block size estimation scheme.

C. Intrinsic Fingerprint Detection Performance

After having found an estimated block size $\hat{B}$, we still have to answer if block processing is truly present. Consider a simple detection problem with the following two hypotheses:

$$H_0 : \mathbf{d}_p = \mathbf{n}, \qquad H_1 : \mathbf{d}_p = \mathbf{s} + \mathbf{n}. \tag{7}$$

Detection of H_1 implies that our block size estimate is correct, while H_0 corresponds to an incorrect block size estimate or an absence of block processing. Actual execution of a likelihood ratio test requires exact knowledge of $\mathbf{s}$ which we do not have. In theory, we can use the signal-to-noise ratio (SNR) as a measure of our detection accuracy. The SNR is defined as $\text{SNR} = P_s/\sigma^2$. Unfortunately, we only have estimates of $\mathbf{s}$ and σ^2 instead of their true values. Calculating the SNR using the signal estimate $\hat{\mathbf{s}}$ is inaccurate since it relies on the accuracy of the signal estimate itself. Furthermore, since $\mathbf{s}$ is not exactly periodic, the variance in the peaks of $\mathbf{s}$ will erroneously contribute to our estimate of the noise power. In practice, we will use the *observed signal-to-noise ratio* (OSNR) for our measure of detection accuracy, defined as

$$\text{OSNR} = \frac{P_{d_p}}{\tilde{\sigma}^2} \tag{8}$$

where $\tilde{\sigma}^2$ is a modified estimate for the noise power which does not include the variance of the peaks in $\mathbf{s}$, i.e,

$$\tilde{\sigma}^2 = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}(d_p(i) - \hat{s}(i;\hat{B}))^2 \tag{9}$$

where the set $\mathcal{I} = \{1,\ldots,M-1\}\backslash\{k\hat{B}|k \in \mathbb{Z}\}$, $\mathcal{I}(i;B) = \{kB + i|k \in \mathbb{Z}\} \cap \{1,\ldots,M-1\}$, and

$$\hat{s}(i;\hat{B}) = \frac{1}{|\mathcal{I}(i;\hat{B})|}\sum_{l\in\mathcal{I}(i;\hat{B})} d_p(l). \tag{10}$$

Fig. 6(a) shows plots of the block size estimation results for the standard test images Lena. We test for block sizes 4, 8, 16, 32, and 64. To create our block-processed images, we use JPEG compression with quality factors from 20 to 90. Each circle represents correct estimation, and each star represents incorrect estimation.

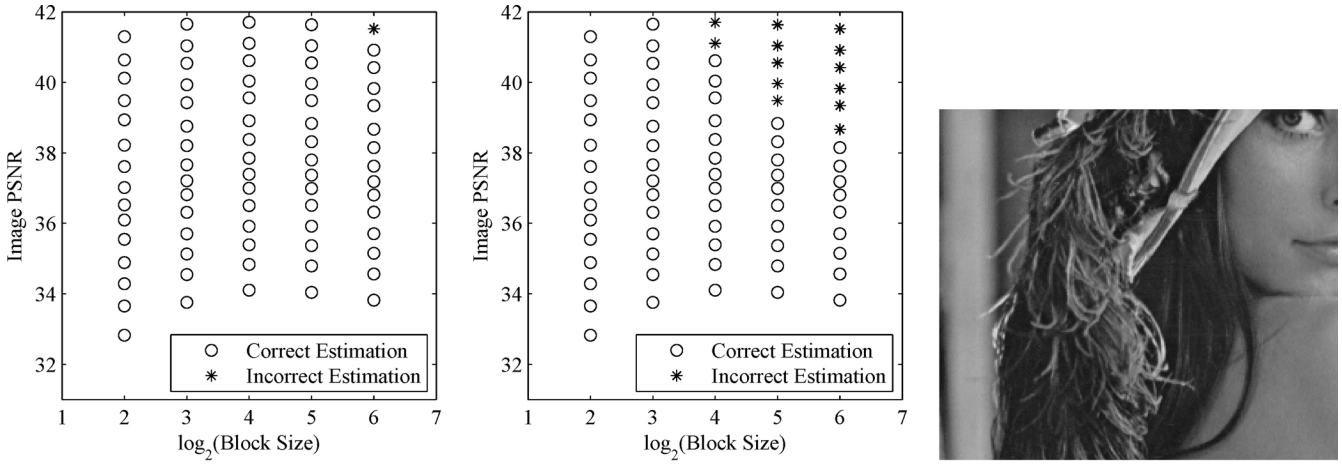
We see that correct estimation varies as a function of both PSNR and block size. Naturally, the strength of block artifacts decreases as image quality increases. As block size increases, the signal $\mathbf{s}$ has fewer periods, and therefore, our estimate $\hat{\mathbf{s}}$ is less accurate. For example, as shown in Fig. 6(a), the estimation accuracy for compressed Lena with block sizes of 4 and 8 in the horizontal direction is 100%, while estimation for block sizes of 16, 32, and 64 fails for PSNR above 41.1, 39.5, and 38.6 dB, respectively. Nevertheless, estimation is still accurate at high PSNRs where the artifacts are not perceptually visible. Fig. 6(b) is a close view of the high frequency part of a JPEG compressed Lena. It is clear that there is no visual blocking effect presence, however, as shown in Fig. 6(a), our block size estimation works perfectly.

We also plot the receiver operating characteristic curve to illustrate our detection accuracy as a function of the OSNR threshold in Fig. 7 which shows the probability of detection P_D versus the probability of false alarm P_F for the test in (7). As the threshold varies, correct estimates with an OSNR that lies above the threshold are hits. Incorrect estimates with an OSNR that lies above the threshold are false alarms. This plot uses detection results from 24 digital images of natural photographs with varying frequency characteristics, all with size 512-by-512. We test for the same block sizes and quality factors mentioned previously and also across both dimensions. We see that our scheme can obtain a P_D of 95.0% for a P_F of 7.4%, and a P_D of 98.0% for a P_F of 16.5%. In practice, we can decrease our OSNR threshold to accommodate a higher P_D. The cost of a miss (i.e., detecting no block artifacts when in fact block processing is present) can be significant in a forensic setting where subsequent forensic tests depend on some block size estimate.

As for the purpose of block size estimation, considering the horizontal and vertical estimations of the block sizes separately can provide a guideline of how good the identification method is. The horizontal and vertical estimations of the block size might not agree. However, the block processing can use rectangle block sizes such as 4×8 and our source-coding forensic detector has no prior information that the preprocessing block is square ($2 \times 2, 4 \times 4, \ldots$) or not. Hence, when the source-coding forensic detector is identifying the block processing, it will estimate the block size horizontally and vertically separately, and combine the estimation results. For instance, if the horizontal block-size estimation is 4 and the vertical estimation is 8, then the source-coding forensic detector will adopt the block size 8×4.

IV. Intrinsic Fingerprint Analysis of Source Encoders

As shown in Fig. 3, after estimating the block size, if any, the forensic detector can start to determine which kind of source encoder used by utilizing the intrinsic fingerprint of each image source encoder as trace of evidence. In this section, we analyze the intrinsic fingerprints of subband coding, transform coding, and differential image coding. For each type of encoder, we investigate the intrinsic fingerprint and define a similarity measure to determine how likely this encoder has been applied.



(a) Vertical dimension (left) and the horizontal dimension (right)

(b) Close view of block effect at PSNR = 38 dB

Fig. 6. Block size estimation results of Lena.

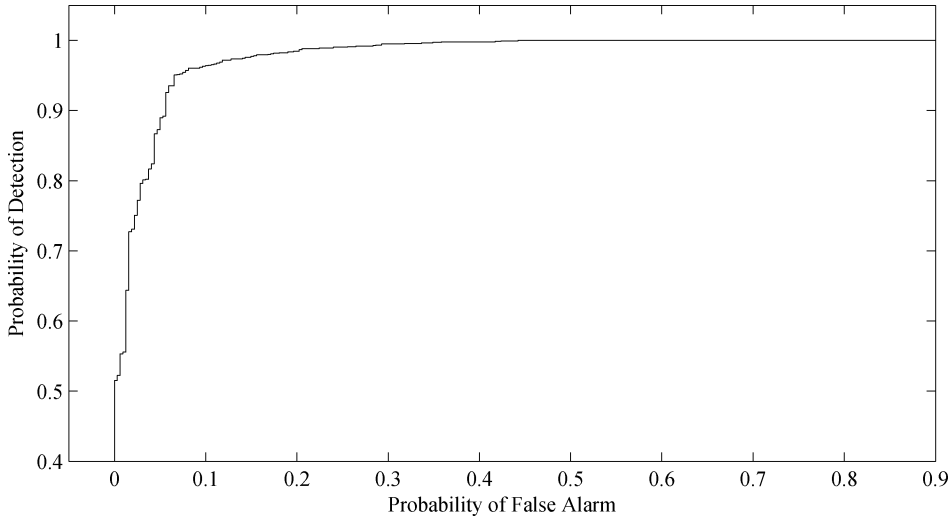


Fig. 7. Performance of the block size estimation algorithm.

A. Transform Coding

In this section, we discuss the most commonly used transforms, DCT and DWT.

1) A Unified Approach for Transform Coding: DCT-based image compression is usually a block-based processing of the image, where the whole image is decomposed into nonoverlapping blocks of the same size (for example 8-by-8 in JPEG baseline) and each block is transformed and compressed separately, while DWT-based source coding schemes often consider the whole image as a single block, apply wavelet transform to the entire image, and decompose it into different frequency bands of different statistics. We unify the transformation mechanisms of DCT coders with that of embedded DWT by drawing some insight from [30], which treated each 8-by-8 block of transform coefficients in a DCT-block coder as a 64-subband decomposition of the original 8-by-8 image block. In other words, we take the (0,0) coefficients of all blocks as a subband, treat all (0,1) coefficients as another subband, and so on.

After tiling all of these subbands together, we obtain a coefficient subband representation similar to the one shown in Fig. 8. In this figure, a discrete cosine transformation with block size of 4 was applied to the original uncompressed image Lena. All of the DCT coefficients of the same frequency are combined into one subband, and these subbands are tiled together. For example, the subband on the first row and first column of

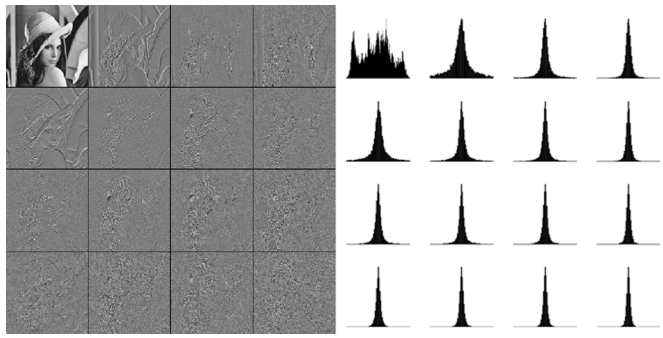


Fig. 8. Left: Reorganization of 4-by-4 DCT coefficients into subbands. Right: Histograms for each coefficient subband.

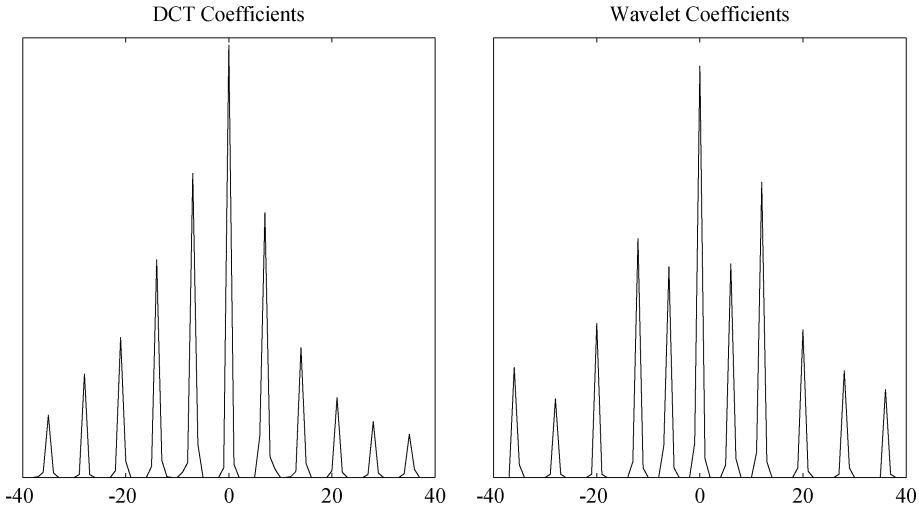


Fig. 9. Example coefficient histograms of two images previously compressed with different schemes. Left: DCT coefficient histogram of position (0,1) after JPEG decoding. Right: Wavelet coefficient histogram of the level-4 LH subband after SPIHT decoding.

the left image is tiled from the DC coefficient, and the subband on the first row and second column is tiled from all the (0,1) DCT coefficients. The corresponding subband histograms are shown in the right figure in Fig. 8. Therefore, we can state that for both types of transforms, if we perform the appropriate subband decomposition and then observe the histogram within each subband, we should find histogram peaks. We can apply the same concept to JPEG2000 images which have been coded using tiling. JPEG2000 allows the use of optional tile sizes of almost any size; subsequent coefficient transformation and quantization is performed on each tile separately. Nevertheless, if we obtain the wavelet coefficient subbands from each tile and then reorganize them as mentioned earlier, we arrive at a single coherent wavelet decomposition.

2) Intrinsic Fingerprint Analysis: First, we consider a DCT block coder. During quantization, the DCT coefficients are discretized. During inverse quantization, the quantized coefficients are multiplied by the quantization step size. As a result, we expect to find peaks in the histogram at multiples of the step size and zeros elsewhere. However, as noted by [31], due to the truncation and rounding affects caused during reconstruction, the histogram peaks do not appear as perfect impulses, as shown in Fig. 9, where Fig. 9(a) shows the DCT coefficient histogram of position (0,1) of JPEG compressed Lena.

Next, consider an embedded DWT coder. Each embedded coder has its own algorithm for deciding the order in which the zerotree is traversed and the coefficients are transmitted. Since the coefficient values are bit-plane encoded, the transform coefficient histogram of the previously compressed image will also contain peaks at the designated reconstruction values, though not necessarily evenly spaced. Fig. 9(b) shows the coefficient histogram at the level-4 LH subband of a SPIHT-coded image with a bit rate of 1.0 bit per pixel. Although quantization is used in all source coding schemes to reduce the bit rate, we could only observe such coefficient histogram peaks in the domain where quantization is applied. For an image that is compressed using subband coding or differential image coding, its DCT or DWT coefficient histograms do not contain such peaks. In addition, for a DCT-compressed image, such peaks do not exist in its DWT coefficient histogram either, and vice versa. Therefore, DCT or DWT coefficient histogram peaks can be used to identify transform-compressed images, as well as to differentiate DCT-compressed images from DWT-compressed ones.

3) Intrinsic Fingerprint Detection: Having discussed a major difference between the transform coefficient histograms of transform-compressed and nontransform compressed images—the presence of histogram peaks, which is the intrinsic fingerprint of transform coding—we would like to characterize this difference between different types of histograms using a distance metric.

Our goal is to compare the transform coefficient histogram for the observed image against the transform coefficient histogram of nontransform compressed image (i.e., no quantization in the transform domain), and decide how similar the two histograms are. Although the original coefficient histogram is irretrievable, we can approximate the original coefficient histogram using a least-squares approximation. Research has previously shown that both the histograms of DCT coefficients and wavelet coefficients of the transform coded image can be accurately modeled using a generalized Gaussian distribution [32]–[34]. Therefore, let $p_i(k)$ be the probability mass function of the original coefficients within a subband, modeled as follows for simplicity:

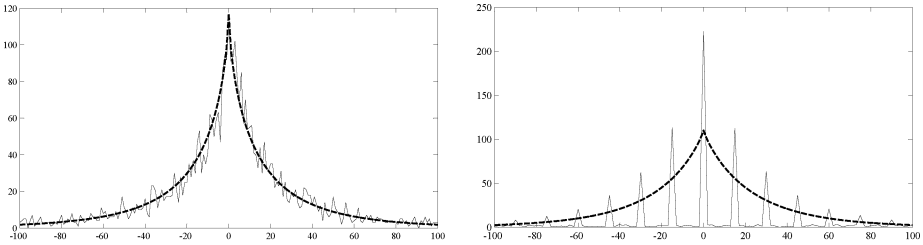
$$p_i(k) = \gamma_i \exp(\lambda_i |k|^{\nu_i}) \tag{11}$$

where i is the index of the subband, γ_i is a normalization constant, $\lambda_i < 0$, and ν_i is the same exponent found in the generalized Gaussian distribution.

Given a fixed ν, linear least-squares can be used to solve for γ and λ. However, this procedure is sensitive to small perturbations in the input data, and it cannot optimize all three parameters simultaneously. To overcome these problems, we will use a nonlinear least-squares method to obtain the best fit. The optimization problem is formulated as follows:

$$\min_{\gamma,\lambda,\nu} \sum_k \left(p(k) - \gamma \exp(\lambda |k|^{\nu})\right)^2, \tag{12}$$

$$\text{s.t.} \gamma > 0, \lambda < 0, \nu > 0. \tag{13}$$



(a) DPCM encoded image (b) DCT-compressed image with step size of 15

Fig. 10. Coefficient histogram of DPCM-encoded and DCT-encoded image, along with the nonlinear least-squares curve fit.

We use a modified Newton method [35] which calculates the Newton direction using a modified Hessian which approaches the true Hessian, and we use a backtracking line search. Furthermore, we incorporate the linear equality constraints on γ, λ, and ν into the optimization problem by using the log-barrier function

$$\min_{\gamma,\lambda,\nu} \sum_{k} (p(k) - \gamma \exp(\lambda |k|^{\nu}))^2 - \mu(\log(\gamma) + \log(-\lambda) + \log(\nu)) \tag{14}$$

where μ is some regularization parameter. This method of exponential curve fitting is relatively fast and offers a very high-quality model of coefficient histograms. Fig. 10(a) illustrates the performance of this estimate for an ac coefficient histogram with a bin width of 0.5 from a differential pulse code modulation (DPCM)-compressed image transformed using the DCT, while Fig. 10(b) illustrates the same fitting method applied to an ac coefficient histogram after DCT-compression has been performed. It is clear that the distance between the true histogram and the least-squares fit is much larger in Fig. 10(b) than in Fig. 10(a).

Now we can calculate how likely the image was coded using the transform coder by utilizing the fact that the least-square fit smooth reconstruction of the histogram will have less correlation with the original histogram if the image is quantized in the transform domain. Hence, we first calculate the correlation Cr_i between the original histogram and the smooth least-square-fit reconstructed histogram for the ith subband: let $\hat{p}_i(k)$ be the histogram of the coefficients in subband i from the reconstructed image and $p_i(k)$ be the least-square fit

$$Cr_i = \frac{||\hat{p}_i(k) * p_i(k)||}{\sqrt{||\hat{p}_i(k)|| * ||p_i(k)||}}. \tag{15}$$

Intuitively, (15) represents the cosine value of the angle between the vectors $\hat{p}_i$ and p_i.

Note that if the least-square curve fits well with the histogram of the received image, then the image is less likely to be quantized in the transform domain. In other words, the image which we are testing is mostly not compressed by the transform encoder. We define the similarity measure for transform coding in the ith subband M_i to measure how likely the image is compressed using transform coding

$$M_i = 1 - \rho_i. \tag{16}$$

If M_i is 0, which means $Cr_i = 1$ thus the histogram in subband i is the same as the histogram of the image that is not compressed using transform encoder. In other words, the image which we are testing is less similar with the image that has passed through the transform encoder. We compute M_i for all coefficient subbands for which sufficient information exists. Our final similarity measure $M_{\text{transform}}$ is the median of all of these M_i values in the case of block transforms, or a weighted mean in the case of wavelet transforms. For the wavelet transform, we weight the similarity value for each subband by the size of the subband before averaging. This weighting guarantees equal contributions from all frequencies in the final similarity measure.

The complete transform method identification algorithm is summarized as follows.

1) Choose a transform to test (e.g., DCT, DWT with Haar basis, DWT with 9/7 basis, etc.). Transform the image. Obtain the subband representation of the coefficients.
2) For each coefficient subband, obtain the histogram. (If insufficient information exists, move on to the next subband.)
3) Approximate the histogram of the original, unquantized coefficients using nonlinear least-squares estimation.
4) Calculate the similarity measure based on the correlation between the observed and the approximated original histogram.
5) Take the median value (for block transforms) or weighted mean (for wavelet transforms) of the similarity measures from all subbands for which sufficient information exists. This value is the final similarity measure. If this value is high, then the transform method tested is the one used during compression.

Before successfully decomposing the image into subbands, the DWT filter lengths and coefficients should be known to the forensic detector. Fortunately, there are only few DWT filter banks in the common DWT-based source encoders such as JPEG 2000 and SPIHT. Hence the forensic detector can examine over all possible DWT filter banks.

4) Intrinsic Fingerprint Detection Performance: First, we show the effectiveness of our method in discriminating between images which have been compressed using transform-based encoders versus images which have been compressed via nontransform based encoders. In Fig. 11, the four starred plots show the similarity measure for four different images, Lena, Baboon, Peppers, and Pentagon, which have been compressed using JPEG (i.e., a DCT with block size of 8×8) for quality factors between 40 and 97. The four triangled plots at the bottom represent the similarity measure for images which have gone through nontransform-based compression. Here we have tested over other encoders studied in this paper, such as subband and DPCM, and both show similar trend. Hence in Fig. 11, we use the H.264 intraprediction encoder. Note that there is a clear distinction between the two sets of lines, even for a quality factor as high as 97. This separation is achieved

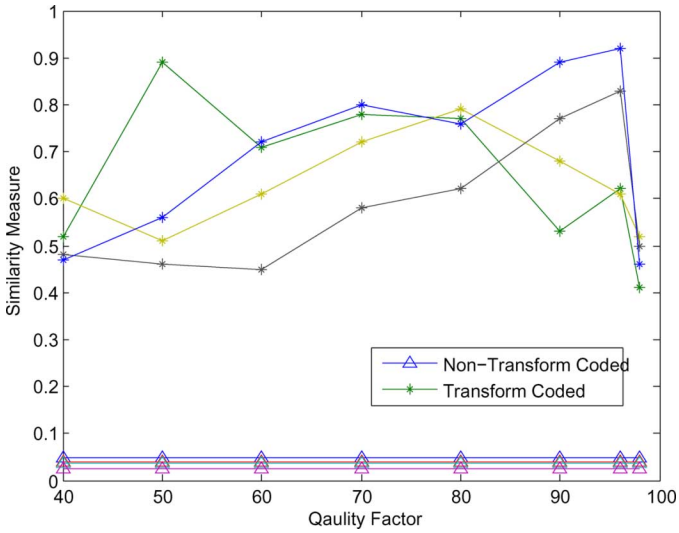


Fig. 11. Similarity measure as a function of quality factor for test images when using transform encoder or nontransform encoders.

thanks to the quality of the nonlinear least-squares estimate of the original coefficient histograms. When least-squares estimation tries to approximate a histogram of coefficients from a transform-compressed image using a generalized Gaussian distribution, the curve fit will be poor and the difference between two histograms will be high, as shown earlier in Fig. 10(b). Also, the difference between the JPEG-compressed images and images that were compressed by other source encoders does not decrease as the quality factor increases. As long as the image was quantized in the transform domain, its coefficient histogram will be like Fig. 10(b) independent of the quantization table is used. Although a higher quality factor implies a smaller quantization step, the coefficient histogram of the compressed image still contains noticeable spikes and is very different from the least-square estimated histogram.

B. Subband Coding

1) Intrinsic Fingerprint Analysis: The subband encoder first passes the image through a bank of filters, called the analysis filter bank, and then applies decimation (downsampling) to decompose the image into several frequency bands of different statistics and different perceptual importance. Then, each frequency band is encoded independently using one of several encoding schemes (for example, differential image coding or DCT coding). At the decoder's side, after decoding each frequency band, the decoded values are upsampled and passed through a bank of synthesis filters to generate the final reconstructed image. There are four sources that will leave traces in a subband-coded image: lack of perfect reconstruction, aliasing, quantization, and signal ringing effect.

When choosing the filter banks, longer filters are often preferred to minimize the aliasing in the later decimation step. However, longer filters also produce more serious ringing effects which are the spurious oscillations in the vicinity of major edges. When addressing the tradeoff between aliasing and ringing, most subband coding schemes use longer filters since aliasing is much less desired than the ringing effect. In addition, ringing exists in all kinds of subband coding no matter which encoding schemes are used to compress each frequency band, and it makes ringing the ideal candidate as the intrinsic fingerprint for all subband coding schemes. DWT coding can be viewed as a special case of subband coding, so it also leaves the same ringing trace in the decoded image. However, in recent development of image source encoders, DWT encoders forms a special and widely used family with standard wavelet bases, so we consider DWT encoders separately using the more significant intrinsic fingerprints as discussed in Section IV-A.II.

2) Similarity Measure: From the previous analysis, the ringing effect is the most significant trace of all subband coding schemes. To extract the trace of ringing effect, we first apply a deringing algorithm to the input image $\mathbf{S}$ to generate a deringed image $\mathbf{S}_r$. The deringing algorithm is maximum-likelihood based which fits the major edges to flat surface model, as in [36]. We then subtract the deringed image $\mathbf{S}_r$ from the test image $\mathbf{S}$ and calculate the difference image $\mathbf{S}_d = \mathbf{S} - \mathbf{S}_r$. The ringing effects often happen on the edges of the image; therefore, if $\mathbf{S}$ is a subband-encoded image, then the energy of the difference image $\mathbf{S}_d$ should concentrate on the edges. If $\mathbf{S}$ was compressed using other schemes, then we would observe that the energy of $\mathbf{S}_d$ is evenly distributed all over the entire image. Fig. 12 shows an example of extracting the trace of ringing effect. Fig. 12(b) shows the difference image $\mathbf{S}_d = \mathbf{S} - \mathbf{S}_r$. If we compare $\mathbf{S}_d$ with the edge map $\mathbf{S}_e$ shown in Fig. 12(a), we can see that the energy of $\mathbf{S}_d$ concentrates on the edges. Fig. 12(c) is the difference image $\mathbf{S}_d = \mathbf{S} - \mathbf{S}_r$ of a DPCM-coded Lena with the same SNR, 36 dB. It is clear that the energy of Fig. 12(c) does not concentrate on the edges, which shows the observed phenomenon used to classify subband coding does not happen in other coding schemes.

To quantify the energy distribution of $\mathbf{S}_d$ on edges, we define the similarity measure of the subband coding schemes as

$$M_{\text{subband}} = \frac{\|(S - S_r) * S_e\|}{\sqrt{N_e \|S - S_r\|}}. \tag{17}$$

In (17), S_e is the binary edge map of S, where a pixel equals 1 if it is detected as an edge pixel and has value 0 otherwise. N_e is the number of pixels that are detected as edge pixels. In (17), the numerator calculates the energy of the edge pixels in the difference image $\mathbf{S} - \mathbf{S}_r$, and the denominator is the normalization term for fair comparison with other similarity measurements. A larger value of M_{subband} gives us a higher confidence that the test image $\mathbf{S}$ was subband coded.

Fig. 13(a) shows the simulation results on the similarity measures of four test images that are encoded using different source coding schemes. The lines that are marked with triangles are the similarity measures of the images compressed by subband encoders, and the starred lines are the similarity measures of those compressed using other encoders (DCT and DPCM). It is obvious that there is a significant gap between the similarity measure values of these two groups, and we can easily distinguish subband-coded images from nonsubband-coded images. The trace of subband coding is stronger when the test image has lower PSNR when compared with the original uncompressed image.

The typical subband coding will further encode the lowest frequency subband (the LL band) using other image coding methods, and the trace of the LL band encoder can also be extracted from the image. In our system, we include two LL-band

Fig. 12. Example of similarity measure calculation of subband coding.

encoders: the DCT encoder and the differential encoder as shown in Fig. 2. We will apply the analysis in Sections IV-A and IV-C based on S_r to give the similarity measure of the two subcategories in the subband category.

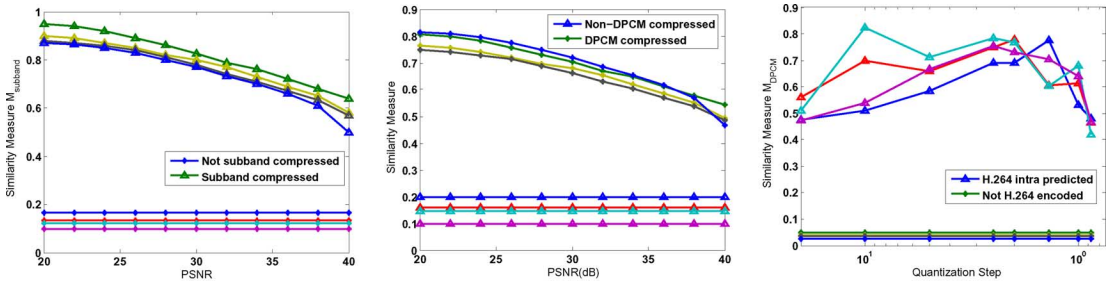
C. Intrinsic Fingerprint of Differential Image Coding

Although differential image coding (DPCM) is one of the very first image encoding methods, it is still applied to current image and video coding standards. The basic idea of differential image coding is to remove the strong correlation between adjacent pixels by representing a pixel as a linear combination of its neighbors and then quantize the residue in the spatial domain.

Today, the most popular lossy application of DPCM is the intraprediction mode of H.264 [28]. H.264 mainly uses DCT to remove the pixel redundancy and the intraprediction in spatial domain as an assistant tool. As a result, the fingerprints of H.264 must be different from that of the traditional differential image coding since they use different tools to remove the pixel redundancy. Therefore, in this section, we propose two different forensic methods to identify traditional differential coding schemes and the H.264 intraprediction.

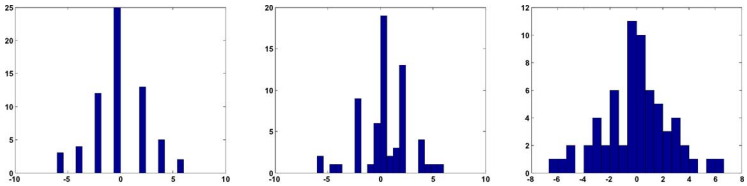
1) H.264 Intraprediction: Prior to encoding, the luma intraprediction in H.264 first subtracts some reference samples from previously encoded and reconstructed blocks. Then, a prediction block (either 4×4 or 16×16) is formed based on the reference samples. There are nine modes for 4×4 blocks and four modes for 16×16 macroblocks [37]. The residue is transformed by integer DCT bases followed by a uniform scalar quantization. The quantization step is the same for all coefficients within the same macroblock. If necessary, the decoded image will go through a deblocking filter which only effects the pixels on the block boundaries.

- *Intrinsic Fingerprint Analysis*: Since residues are quantized in the integer DCT domain, we expect to observe the same histogram peaks as transform coding if we calculate the residue correctly. However, the boundary pixels of the decoded image might go through linear filters to reduce the blocking effect, thus changing the value of reference samples in the decoded image. As a result, the peaks do not appear as perfect impulses, as shown in Fig. 14. Since the value of the boundary pixels might be changed by deblocking filters, the residue of the boundary pixels are not included in the histogram calculation. Fig. 14 shows the histogram in transform domain of the 64 residues in a 16×16 macroblock under the 4×4 intraprediction mode 0. The boundary pixels in Fig. 14(a) are not deblocking filtered, thus Fig. 14(a) shows perfect peaks. Fig. 14(b) is the histogram of the transformed residue pixels after the whole H.264 reconstruction process (including deblocking). It is clear that the peaks are not perfect impulses but still much higher than other values, in comparison to the Fig. 14(c), which is calculated from a JPEG-compressed image.
- *Similarity Measure*: Since the intrinsic fingerprint of the H.264 intraprediction is the presence of histogram peaks in transform domain, we would like to characterize the difference between different types of histograms using a distance metric as in Section IV-A3.
  We first need to obtain the estimated residues of each block (or macroblock). In the encoding process of H.264 intraprediction, the prediction mode is chosen to minimize the sum of residue energy. Therefore, we apply the same algorithm to estimate the prediction mode by choosing the one that has minimal square error.
  The similarity measure of the macroblock i is defined and calculated as in (16). We calculate the similarity measure M_i over all the macroblocks of the whole image, and choose the median value as the similarity measure of differential image coding M_{H264}.
  We test on four images, Lena, Baboon, Peppers, and Pentagon, that are compressed using H.264 intraprediction mode with macroblock 16-by-16 and prediction mode 0. The similarity measures M_{H264} versus quantization step are shown in Fig. 13(c). The lines marked with stars are for H.264-compressed images, and the lines with triangle markers represent those transform-coded and subband-coded images. From Fig. 13(c), it is obvious that there is a significant gap between M_{H264} of these two groups. Therefore, M_{H264} can accurately identify H.264-coded images from images compressed using other source coding schemes studied in this paper. Note that the similarity measure does not decrease significantly when the quantization step decreases. Even with small quantization step sizes, the residues are still discretized. When the quantization step size becomes very small (for instance,



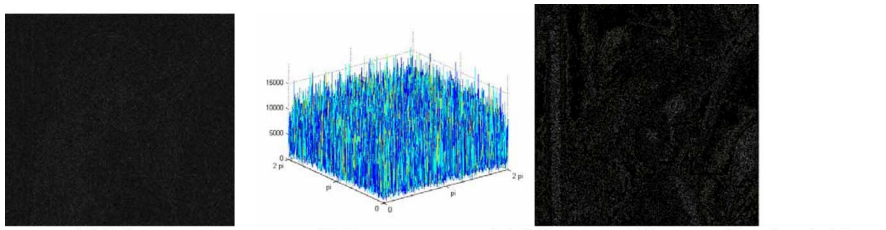
(a) Subband and Non-subband compressed images (b) DPCM and Non-DPCM compressed images (c) H.264 intra predicted and Non-H.264 intra predicted images

Fig. 13. Similarity measures of subband coding, differential coding, and H.264 intraprediction on test images.



(a) Not deblocking filtered (b) Deblocking filtered (c) Not Intra-prediction

Fig. 14. Example of residue histogram of 4×4 intraprediction in H.264.



(a) S-S_d (b) F_d (c) S-S_d extracted from a subband-coded Lena

Fig. 15. Example of intrinsic fingerprints of differential coding at SNR $=$ 30 dB.

0.875), the decoded image is not deblocked, which keeps the sharp peaks in the histogram. Therefore, the intrinsic fingerprint is still preserved.

2) Traditional Differential Coding: In traditional differential image coding schemes, the encoders are allow to change the estimation coefficients and the filter lengths. As a result, the estimation mode is not a small finite set as in H.264 intraprediction, thus the residue histogram cannot be estimated as in Section IV-C.

- *Intrinsic Fingerprint Analysis*: Assuming that the received image $\mathbf{S}$ was differential-coded, to identify the intrinsic fingerprints, we first apply image denoising [38] to remove the white quantization noise. We then use the difference between the test image $\mathbf{S}$ and the denoised image $\mathbf{S}_d$ to estimate the quantization noise in DPCM. If the received image $\mathbf{S}$ was compressed using differential coding, then the estimated quantization noise $\mathbf{S} - \mathbf{S}_d$ should have a white spectrum.

  Fig. 15 shows an example where the Lena image was encoded using DPCM with PSNR of 30 dB. Fig. 15(a) and (b) show the estimated quantization noise $\mathbf{S} - \mathbf{S}_d$ and its spectrum, respectively. As we can see, $\mathbf{S} - \mathbf{S}_d$ has a flat spectrum, resembling a white noise. As an illustration of the intrinsic fingerprints of DPCM encoders, in Fig. 15(c) we also show the $\mathbf{S} - \mathbf{S}_d$ of a subband coded Lena with the same SNR 30 dB; it is clear that Fig. 15(c) does not show the same characteristic as Fig. 15(a).
- *Similarity Measure*: Let F_d be the spectrum of $\mathbf{S} - \mathbf{S_d}$ and F be the spectrum of a white noise with the same power as F_d. To quantify the resemblance of F_d to F, we define the similarity measure M_{DPCM} for DPCM as

$$M_{\mathrm{DPCM}} = \frac{||F_d * F||}{\sqrt{||F|| * ||F_d||}}. \tag{18}$$

  Similar to (17), the numerator calculates the similarity between F_d and F and the denominator normalizes the similarity for fair comparison with other measures.

  We test on four images that are compressed using different coding schemes and plot their M_{DPCM} in Fig. 13(b). The lines marked with stars are the similarity measures of DPCM-compressed images, and the lines with triangle markers represent those transform-coded and subband-coded images. From Fig. 13(b), it is obvious that there is a significant gap between M_{DPCM} of these two groups, and that M_{DPCM} can accurately differentiate DPCM-coded images from images compressed using other source coding schemes.

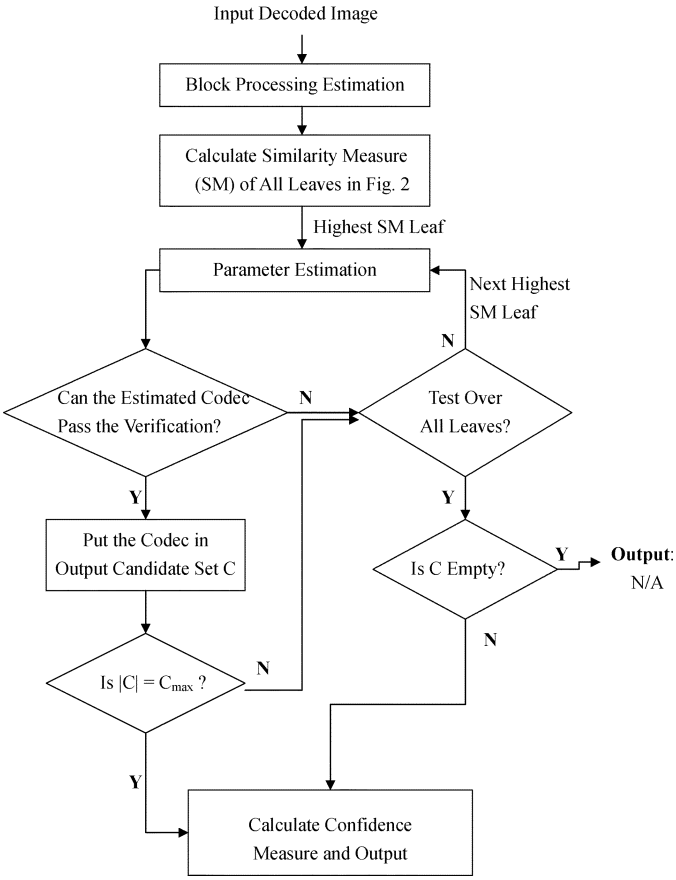


Fig. 16. Detailed image source-coding forensic system flowchart.

V. Image Source Coding Forensic Detector

Now that we have all the similarity measures for different source coding schemes in Fig. 2, we can construct an image coding forensic system. The system flowchart is shown in Fig. 16, where $0 \leq \mathrm{TH} \leq 1$ is a predetermined threshold and $C_{\max}$ is the maximum size of the output candidate set determined by the system designer. The threshold TH is determined by a training database including 24 natural images, and these images are not included in the test database.

The key steps to detect the source coding schemes are as follows:

1) Determine whether the image has been block processed, if so, estimate the block size.
2) Apply the intrinsic fingerprint detection as discussed in Section IV, and calculate the similarity measure of all leaves in Fig. 2.
3) Choose the node in the tree with the highest similarity measure and estimate the coding parameters.
4) Calculate the similarity K between the re-encoded image and the received image.
5) If $K <= TH$, discard this source coding scheme. If there are still some other source encoders in the tree, go back to step 3. If there are no more source encoders in the tree, and the output candidate set $\mathbf{C}$ is empty, return N/A. If it is not empty, compute the confidence measure and output.
6) If $K > TH$, put this codec in the output candidate set $\mathbf{C}$.
7) Test whether the size of the output candidate set equals $C_{\max}$. If so, compute the confidence measure and output the one with lowest noise variance as calculated in Section V-B. If not, go back to Step 3.

A. Parameter Estimation

For DCT-based transform coding, we take JPEG as an example. The DCT basis can be directly acquired once we finish the block processing detection as discussed in Section III-A, so all we need to discuss here is quantization table estimation.

First, given a block size of B, we obtain the histogram for each of the B^2 transform coefficient positions by counting the value for one particular coefficient position over all blocks and finding the distance between two consecutive peaks in the histogram to determine the quantization step size [39]. However, high-frequency coefficients do not contain enough information to determine the step size. Therefore, we extrapolate the remaining coefficients by expressing the quantization table as a linear combination of existing quantization tables.

To do this, we find all of the coefficients which were accurately estimated in our initial procedure. Let $\mathbf{y} \in \mathbb{Z}^{B^2}$ be the vector of quantization table elements estimated by finding the distance between consecutive peaks in the coefficient histograms. Then let $\tilde{\mathbf{y}}$ equal the vector $\mathbf{y}$ without those coefficients whose step size has not yet been determined. Let the matrix $\mathbf{A} = [\mathbf{a}_1 \quad \cdots \quad \mathbf{a}_K]$ be a collection of existing quantization tables, and let $\tilde{\mathbf{A}}$ only include the same coefficient positions as in $\tilde{\mathbf{y}}$. In order to minimize $\|\tilde{\mathbf{y}} - \sum z_k \tilde{\mathbf{a}}_k\|$, for some scalars z_k, we solve the following normal equations:

$$\tilde{\mathbf{A}}^T \tilde{\mathbf{A}} \mathbf{z} = \tilde{\mathbf{A}}^T \tilde{\mathbf{y}} \tag{19}$$

which gives us $\mathbf{z}$. The rest of the coefficients are then estimated using $\mathbf{y} = \mathbf{A}\mathbf{z}$.

To illustrate the accuracy of this approach, we ran 1000 trials of the following experiment. We define the matrix $\mathbf{A}$ to include ten existing quantization tables including those used in Adobe Photoshop, digital cameras, and the JPEG standard luminance and chrominance tables [40]. We compressed a 512-by-512 test image using an 8-by-8 quantization table Q obtained through a random linear combination of 23 existing tables. Fig. 17 illustrates the average absolute error obtained using our quantization table estimation method for each of the 64 coefficient frequencies.

As we see from Fig. 17, the coefficients in the low- and mid-frequency ranges are estimated perfectly. For high-frequency coefficients, the algorithm produces estimates that are close to the original table elements. The accuracy of the algorithm certainly depends on how similar the tested quantization table is to the set of known tables. Clearly, if the quantization table used during compression happens to be an element in the set of known quantization tables which comprise the training data, then the quantization table estimate will be perfect.

1) Subband Coding: For subband coding schemes, the encoding parameters that we need to estimate are the filter banks and the decimation factor (or equivalently, the number of frequency bands).

Since the quadrature mirror filter (QMF) banks are the most commonly used filter banks in the literature, we consider some of the most popular QMFs [27] in our system. To estimate the QMF that was used, for each candidate, we first apply a subband coder of level 1 to the test image and divide it into four subbands

$$E[|\hat{Q} \leftarrow Q|] = \begin{bmatrix} 0.078 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.002 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.006 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.001 & 0.145 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.073 & 0.131 \\ 0 & 0 & 0 & 0 & 0 & 0.066 & 0.170 & 0.297 \\ 0 & 0 & 0.009 & 0.005 & 0.084 & 0.195 & 0.275 & 0.453 \\ 0.023 & 0.177 & 0.168 & 0.118 & 0.148 & 0.326 & 0.475 & 0.737 \end{bmatrix} \quad (20)$$

Fig. 17. Average absolute error for the quantization table estimation method.

(LL, LH, HL, and HH) in the 2-D spatial domain. We choose the QMF that gives the most energy concentration in LL subband.

2) Differential Image Coding:

Traditional DPCM: Without loss of generality, we assume there are n possible lengths of the spatial filters $L = [L_1, L_2, \ldots, L_u]$. Let $[x_L, x_H]$ be the residue range. If the prediction error of the current pixel is larger than x_L, increase the filter length to the next level; if the prediction error is less than x_L, shorten the filter length.

- First, divide the image $\mathbf{S}$ into flat regions and nonflat regions by the edge map.
- Compute the energy of $\mathbf{S} - \mathbf{S_d}$, the estimated quantization error, in both the flat regions and the nonflat region. If the two values are not comparable, then a nonadaptive DPCM encoder was used.
- To determine the order of the spatial filter, and the filter coefficients jointly, for each filter length n, solve the least-square solution for the filter coefficients $\mathbf{A}_n$ and the corresponding error $||\mathbf{S} * \mathbf{A}_n - \mathbf{S}||^2$ for different filter lengthes. The estimate filter length n_{DPCM} and filter coefficient $A_{n_{\mathrm{DPCM}}}$ is the one that minimizes $|||\mathbf{S} - \mathbf{S_d}||^2 - ||\mathbf{S} * \mathbf{A}_n - \mathbf{S}||^2|$

H.264 Intraprediction: The parameters of H.264 intraprediction are the prediction mode and the quantization step of each macroblock. The prediction mode is already identified in the similarity measure calculation as in Section IV-C, thus the only parameter to be estimated is the quantization step for each macroblock.

- For each macroblock i, threshold the histogram p_i with its least-square fit $\hat{p}_i$. If $p_i(k) \geq \hat{p}_i$, then $p'_i(k) = p_i(k) - \hat{p}_i$; otherwise $p'_i(k) = 0$
- Given a quantization step q in H.264 standard, generate an impulse train in transform domain with interval q.
- Multiply the thresholded histogram p'_i with the impulse train, and calculate the energy.
- Choose the quantization step that gives the highest energy.

B. System Confidence Measure

In our system, we also design a confidence measure to quantify the confidence of the proposed source-coding forensic detector on the classification and estimation results. A higher confidence value in estimation would increase the trustworthiness of the decision made by a forensic analyst.

Given a test image $\mathbf{S}$, for the ith source-coding scheme candidate, we first remove from $\mathbf{S}$ the effect of compression and use the resulting $\mathbf{S}_r$ image to estimate the original uncompressed image. For example, with subband coding, we use deringing to remove the ringing effect and thus estimate the original image, and we apply denoising to the test image to remove the white quantization noise with differential image coding. Then, we use the ith compression scheme to re-encode $\mathbf{S}_r^{(i)}$ and generate the re-encoded image $\mathbf{S}_d$. We then calculate the difference between the test image and the re-encoded image, i.e., $\mathbf{e}^{(i)} = \mathbf{S} - \mathbf{S}_d$.

Let $\sigma_e^2(i)$ denote the variance of $\mathbf{e}^{(i)}$. If the ith candidate is the source coding scheme that was used to generate the test image $\mathbf{S}$, then $\mathbf{S}_r^{(i)}$ will be the same as $\mathbf{S}$ except for rounding errors. If we treat $\mathbf{e}^{(i)}$ as noise, then $\mathbf{e}^{(i)}$ should have low energy, and $\sigma_e^2(i)$ will have a much smaller value than other $\sigma_e^2(j)$, where $j \neq i$. Let C be the number of candidate source coding schemes, and let $\sigma^2_{\min}$ denote the minimum of $\{\sigma_e^2(1), \sigma_e^2(2), \ldots, \sigma_e^2(C)\}$. The smaller that $\sigma^2_{\min}$ is, the more confidence we have in the estimation results.

We propose a noise-variance-based measure to quantify the confidence level on the estimation result. Define $P = [P_1, P_2, \ldots, P_C]$, where

$$P_i = \frac{\dfrac{1}{\sigma_e^2(i)}}{\displaystyle\sum_{i=1}^{C} \frac{1}{\sigma_e^2(i)}} \quad (21)$$

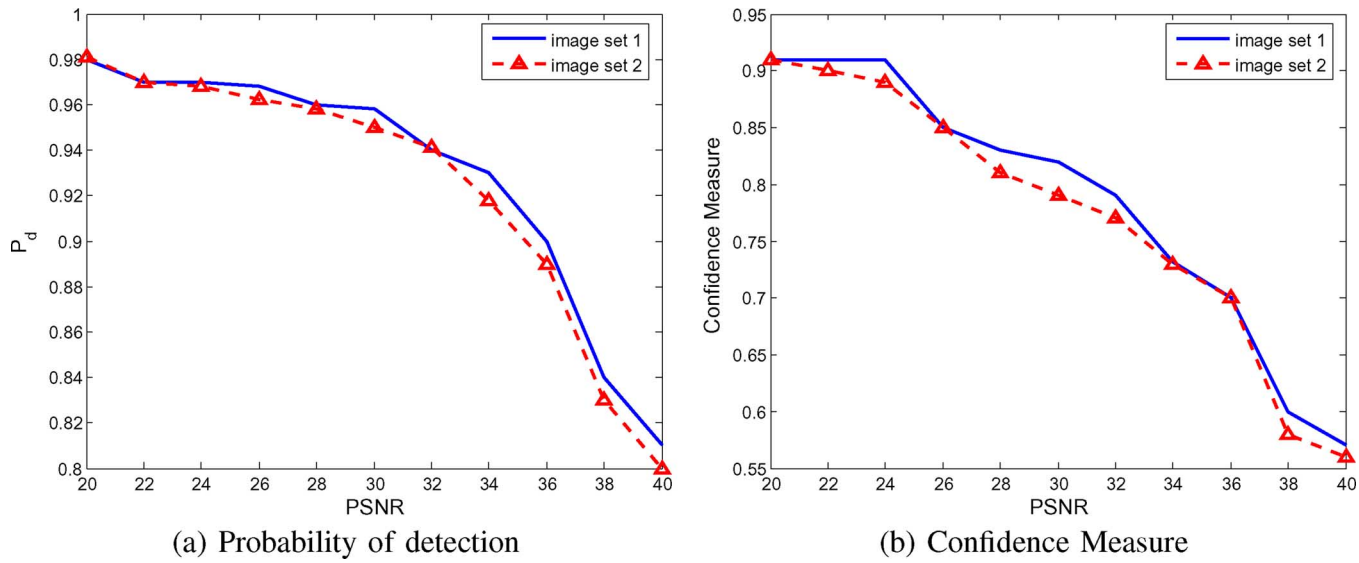
is the normalized variance reciprocal vector. Our confidence measure M_{all} is defined as

$$M_{\mathrm{all}} = 1 - \frac{H(P)}{\log_2 C} \quad (22)$$

where $H(P) = \Sigma_{i=1}^{c} P_i \log_2(1/P_i)$ is the entropy of the vector P. From the above analysis, when $\sigma^2_{\min}$ is much smaller than any other $\sigma_e^2(j)$ (or equivalently, $1/\sigma^2_{\min}$ is much larger than the reciprocals of all other $\sigma_e^2(j)$), the vector P has a smaller entropy $H(P)$, and we have more confidence in the classification and estimation results.

VI. Simulation Results

In our simulations, we have two sets of test images. For image set 1, we collect seven benchmark images that are widely used in image processing and analysis: Lena, Baboon, Barbara, Couple, Man, Boat, and Tank. We test over five different categories of image encoders as shown in the leaf nodes in Fig. 2, and within every category, we have used different sets of parameters to compress the images, resulting in a database of 427 images. For the DCT, we consider the most common block transform that uses a block size of 8-by-8, 16-by-16, and 4-by-4 and we use baseline JPEG with a quality factor of 60, 70, and 80 to quantize the DCT coefficients. For DWT-based transforms, we con-



(a) Probability of detection (b) Confidence Measure

Fig. 18. Performance of the image source coding forensic detector.

TABLE I
CONFUSION MATRIX WHEN PSNR $=$ 36 dB.

		DCT Encoder	DWT Encoder	Subband Encoder		Differential Image Encoder	N/A
				DCT	DPCM		
DCT Encoder		90.7%	1.1%	2.0%	1.2%	0.2%	4.8%
DWT Encoder		0.8%	90.8%	3.4%	1.9%	0.3%	2.8%
Subband Encoder	DCT	0.3%	1.1%	90.1%	1.2%	0.2%	6.1%
	DPCM	0.1%	1.2%	3.0%	91.3%	0.4%	5.0%
Differential Image Encoder		0.7%	0.6%	0.3%	3.2%	91.1%	4.1%

sider the three widely adopted wavelet transforms, 5/3, 9/7, and 17/11, as an example in our simulations, and we use the standard five-level multiresolution decomposition. SPIHT is used for quantization with a bit rate of 0.8 and 0.9 bits per pixel. For DPCM, we set the set of filter length as [10, 15, 20]. For H.264 intraprediction, we use either 16×16 macroblocks or 4×4 blocks. For subband coding, we use the filters as in [41]. Image set 2 contains 20 raw images taken from two digital cameras (Canon and Olympus) and then sent through the compression and decompression process.

The PSNR of the compressed images varies from 20 to 40 dB. Fig. 18(a) illustrates the probability that our forensic detector outputs the correct category of source encoder (DCT based, DWT based, DCT subband, DPCM subband, DPCM and H.264) of the two image sets. The results of both image sets are similar; both show that our method works quite well with accuracy over 90% when PSNR is less then 36 dB. Not surprisingly, the result degrades when PSNR increases. With a higher PSNR, the difference between the compressed image and the original one is smaller, and therefore, the trace of the source coding schemes is more difficult to detect. Thus, it is more challenging to determine what kind of source encoder has been applied to the test image with a higher PSNR. Fig. 18(b) shows the corresponding confidence measure of the image source coding forensic detector for the two image sets. Though the performance degrades 18% at high PSNR (PSNR $=$ 40 dB), the system also outputs a smaller confidence measure and indicates to the forensic analyst that there might be errors.

Our forensic detector provides very accurate detection (over 90% detection rate) when the PSNR is less than 36 dB. We also provide indicating confidence measure over all PSNR range to give the users of the forensic detector the level of correctness. When the detection rate of the system is lower than 85%, the confidence measure is less than 0.65 (when PSNR $>=$ 38 dB) which reflects the uncertainty of the system. Such results show our image source coding detector is a reliable forensic tool over a wide range of PSNR.

Table I shows the confusion matrix between the four source encoders when PSNR $=$ 36 dB within image set 1. Here, the DCT-subband method and the DPCM-subband method are separated in the subband coding category to show the trend; H.264 and traditional DPCM are combined together to simplify the analysis. Each row represents the true source coding scheme that was used to generate the test image, and each column represents the encoder that our image source coding forensic engine outputs. The forensic engine will output N/A if no candidate image encoders can pass the verification process as in Fig. 16. From Table I, the largest estimation error for every source en-

coder occurs when outputting N/A. That is, the intrinsic fingerprint has low energy in high image quality as PSNR $= 36$ dB. Thus, the forensic engine cannot identify the source encoders. The intrinsic fingerprint we find for each source encoder is distinguishing enough that even when the fingerprint energy is low, the image source encoder forensic detector still does not confuse different encoders within the system.

Another big estimation error occurs when identifying a DWT-compressed image as a subband-encoded image or when misclassifying the two subcategories in the subband encoder family. A DWT-based encoder has a similar encoding structure to a subband coder, and therefore, we can also observe ringing effects in some DWT-compressed images. The DCT-subband method and DPCM-subband method also share the ringing effect as the most significant fingerprint, where the fingerprint of DCT or DPCM has lower significance and causes misclassification.

VII. Conclusion

In this paper, we have proposed a forensic methodology via intrinsic fingerprints to identify the compression method applied, if any, to a digital image, along with estimation of all parameters and a confidence measure of the estimated coding scheme. Our forensic detector does not need any information other than the decoded image at the receiver because we investigate and probe the unique intrinsic fingerprint of image source encoder embedded in the received image.

By using the intrinsic fingerprint of each source encoder, our image source coding forensic detector via intrinsic fingerprint can choose the correct image encoder among transform-based encoders, subband encoders, and DPCM encoders with probability higher than 90% when PSNR ≤ 36 dB. Even with PSNR of 40 dB, the probability of correct estimation is still 80%. In the future, as mentioned in Section IV-A, our system can be applied to JPEG 2000, KL-transform based encoders, and other lossy image source encoders.

In our future work, we plan to investigate multiple image source coding forensics. Since many digital cameras in the market do not provide raw sensor output, and since most digital images are compressed before any possible tampering, detecting how many times the images have been compressed and what the source encoders are will provide valuable information on identifying the datapath that the image has gone through, for instance, the order of compression steps performed previously.

References

[1] G. Friedman, "The trustworthy digital camera: Restoring credibility to the photographic images," *IEEE Trans. Consumer Electron.*, vol. 39, no. 4, pp. 905–910, Nov. 1993.

[2] K. J. R. Liu, W. Trappe, Z. J. Wang, M. Wu, and H. Zhao, *Multimedia Fingerprinting Forensics for Traitor Tracing*. New York: Hindawi Pub. Corp., 2005, EURASIP Book Series on Signal Processing and Communications.

[3] H. Farid, Digital image ballistics from JPEG quantization Tech. Rep..

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

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

[6] S. Liu and A. C. Bovik, "Efficient DCT-domain blind measurement and reduction of blocking artifacts," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 12, pp. 1139–1149, Dec. 2002.

[7] W. Gao, C. Mermer, and Y. Kim, "A de-blocking algorithm and a blockiness metric for highly compressed images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 12, pp. 1150–1159, Dec. 2002.

[8] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pt. 2, pp. 758–767, Feb. 2005.

[9] A. C. Popescu and H. Farid, "Statistical tools for digital forensics," in *Int. Workshop Information Hiding*, Toronto, Canada, 2004.

[10] J. Fridrich, D. Soukal, and J. Lukas, "Detection of copy-move forgery in digital images," in *Proc. Digital Forensics Research Workshop*, Columbus, OH, Aug. 2003.

[11] Z. Fan and R. L. de Queiroz, "Identification of bitmap compression history: Jpeg detection and quantizer estimation," *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 230–235, Feb. 2003.

[12] J. Lukas and J. Fridrich, "Estimation of primary quantization matrix in double compressed JPEG images," in *Proc. Digital Forensics Research Workshop (DFRWS 2003)*, Cleveland, OH, Aug. 2003.

[13] T.-T. Ng, S.-F. Chang, and Q. Sun, "Blind detection of photomontage using higher order statistics," in *IEEE Int. Symp. Circuits and Systems (ISCAS)*, Vancouver, Canada, May 2004.

[14] H. Farid, "Blind inverse gamma correction," *IEEE Trans. Image Process,*, vol. 10, no. 10, pp. 1428–1433, Oct. 2001.

[15] H. Farid and A. Popescu, "Blind removal of image non-linearities," in *Proc. IEEE Int. Conf. Computer Vision*, Jul. 2001, vol. 1, pp. 76–81.

[16] M. K. Johnson and H. Farid, "Exposing digital forgeries in complex lighting environments," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3, pp. 450–461, Jun. 2007.

[17] Y.-F. Hsu and S.-F. Chang, "Detecting image splicing using geometry invariants and camera characteristics consistency," in *IEEE Int. Conf. Multimedia and Expo (ICME)*, Toronto, Canada, Jul. 2006.

[18] H. Farid and S. Lyu, "Higher-order wavelet statistics and their application to digital forensics," in *IEEE Workshop Statistical Analysis in Computer Vision*, Feb. 2003, p. 94.

[19] T.-T. Ng, S.-F. Chang, and Q. Sun, "Blind identification of photomontage using higher order statistics," in *IEEE Int. Symp. Circuits and Systems*, Vancouver, Canada, May 2004.

[20] Z. J. Geradts, J. Bijhold, M. Kieft, K. Kurosawa, K. Kuroki, and N. Saitoh, "Methods for identification of images acquired with digital cameras," *Proc. SPIE, Enabling Technologies for Law Enforcement and Security*, vol. 4232, pp. 505–512, Feb. 2001.

[21] K. Kurosawa, K. Kuroki, and N. Saitoh, "Ccd fingerprint method – Identification of a video camera from videotaped images," in *Proc. IEEE Int. Conf. Image Processing*, Oct. 1999, vol. 3, pp. 537–540.

[22] M. Kharrazi, H. T. Sencar, and N. Memon, "Blind source camera identification," in *Proc. IEEE Int. Conf. Image Processing*, Oct. 2004, vol. 1, pp. 709–712.

[23] S. Bayram, H. T. Sencar, N. Memon, and I. Avcibas, "Source camera identification based on cfa interpolation," in *Proc. IEEE Int. Conf. Image Processing*, Sep. 2005, vol. 3, pp. 69–72.

[24] A. Swaminathan, M. Wu, and K. J. R. Liu, "Non-intrusive component forensics of visual sensors using output images," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 1, pp. 91–106, Mar. 2007.

[25] Scientific Working Group on Imaging Technology (SWGIT) International Association for Identification, 2006 [Online]. Available: http://www.theiai.org/guidelines/swgit/index.php

[26] R. Veldhuis and M. Breeuwer, *An Introduction to Source Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[27] J. Woods and S. O'Neil, "Subband coding of images," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 5, pp. 1278–1288, Oct. 1986.

[28] I. E. Richardson, *H.264 and MPEG-4*. New York: Wiley, 2003.

[29] J. D. Wise, J. R. Caprio, and T. W. Parks, "Maximum likelihood pitch estimation," *IEEE Trans. ASSP*, vol. ASSP-24, no. 5, pp. 418–423, Oct. 1976.

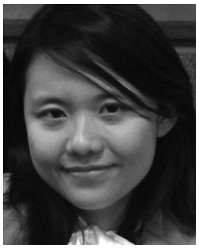
[30] Z. Xiong, O. G. Guleryuz, and M. T. Orchard, "A DCT-based embedded image coder," *IEEE Signal Process. Lett.*, vol. 3, no. 11, pp. 289–290, Nov. 1996.

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

[32] E. Y. Lam and J. W. Goodman, "A mathematical analysis of the DCT coefficients distributions for images," *IEEE Trans. Image Process.*, vol. 9, no. 10, pp. 1661–1666, Oct. 2000.

[33] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.

[34] F. Muller, "Distribution shape of two-dimensional DCT coefficients of natural images," *Electron. Lett.*, vol. 29, pp. 1935–1936, Oct. 1993.
[35] S. G. Nash and A. Sofer, *Linear and Nonlinear Programming*. New York: McGraw-Hill, 1996.
[36] S. Yang, Y.-H. Hu, T. Nguyen, and D. Tull, "Maximum-likelihood parameter estimation for image ringing-artifact removal," *IEEE Circuits Syst. Video Technol.*, vol. 11, no. 8, pp. 963–973, Aug. 2001.
[37] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*. New York: Van Nostrand, 1993.
[38] M. K. Mihcak, I. Kozintsev, K. Ramchandran, and P. Moulin, "Low-complexity image denoising based on statistical modeling of wavelet coefficients," *IEEE Signal Process. Lett.*, vol. 6, no. 12, pp. 300–303, Dec. 1999.
[39] R. Samadani, Characterizing and Estimating Block DCT Image Compression Quantization Parameters Imaging Systems Laboratory, HP Laboratories, Palo Alto, Tech. Rep. HPL-2005–190, 2005.
[40] C. Hass, Impulseadventure – JPEG Quality and Quantization Tables for Digital Cameras, Photoshop 2008 [Online]. Available: http://www.impulseadventure.com/photo/jpeg-quantization.html
[41] J. W. Woods, *Subband Image Coding*. Norwell, MA: Kluwer, 1991.



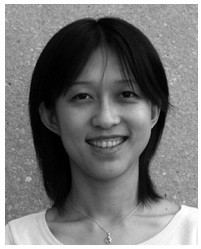
W. Sabrina Lin (M'06) received the B.S. and M.S. degrees in electrical engineering from National Taiwan University in 2002 and 2004, respectively. She is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department, University of Maryland, College Park.

Her research interests are in the area of information security and forensics, multimedia signal processing, and multimedia social network analysis. She received the University of Maryland Future Faculty Fellowship in 2007.

Steven K. Tjoa received the B.S. degree in computer engineering in 2004 and the M.S. degree in electrical engineering in 2006 from the University of Maryland, College Park, where he is currently pursuing the Ph.D. degree in electrical engineering.

Since 2005, he has been a graduate research assistant with the Signals and Information Group at the University of Maryland. His research interests include multimedia signal processing, digital image forensics, and music information retrieval.

Mr. Tjoa earned awards from the University of Maryland including a Presidential Scholarship in 2000, the George Corcoran Memorial Award from the Department of Electrical and Computer Engineering in 2007, a Graduate School Fellowship in 2008, and a Future Faculty Fellowship from the Clark School of Engineering in 2008.



H. Vicky Zhao (S'01–M'04) received the B.S. and M.S. degrees from Tsinghua University, China, in 1997 and 1999, respectively, and the Ph.D. degree from University of Maryland, College Park, in 2004, all in electrical engineering.

She was a Research Associate with the Department of Electrical and Computer Engineering and the Institute for Systems Research, University of Maryland, College Park, from January 2005 to July 2006. Since August 2006, she has been an Assistant Professor with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. Her research interests include information security and forensics, multimedia, digital communications, and signal processing.

Dr. Zhao received the IEEE Signal Processing Society (SPS) 2008 Young Author Best Paper Award. She coauthored the book *Multimedia Fingerprinting Forensics for Traitor Tracing* (Hindawi, 2005). She has been the Associate Editor for IEEE SIGNAL PROCESSING LETTERS since November 2008.

K. J. Ray Liu (S'86–M'90–SM'93–F'03) is a Distinguished Scholar-Teacher of the University of Maryland, College Park. He is Associate Chair of Graduate Studies and Research of the Electrical and Computer Engineering Department and leads the Maryland Signals and Information Group conducting research encompassing broad aspects of information technology including communications and networking, information forensics and security, multimedia signal processing, and biomedical technology. He was Vice President – Publications and will serve as President-Elect in 2010. He was the Editor-in-Chief of *IEEE Signal Processing Magazine* and the founding Editor-in-Chief of *EURASIP Journal on Applied Signal Processing*. His recent books include *Cooperative Communications and Networking* (Cambridge Univ. Press, 2008); *Resource Allocation for Wireless Networks: Basics, Techniques, and Applications* (Cambridge Univ. Press, 2008); *Ultra-Wideband Communication Systems: The Multiband OFDM Approach* (IEEE-Wiley, 2007); *Network-Aware Security for Group Communications* (Springer, 2007); *Multimedia Fingerprinting Forensics for Traitor Tracing* (Hindawi, 2005); and *Handbook on Array Processing and Sensor Networks* (IEEE-Wiley, 2009).

Dr. Liu is the recipient of numerous honors and awards including best paper awards from IEEE Signal Processing Society, IEEE Vehicular Technology Society, and EURASIP; IEEE Signal Processing Society Distinguished Lecturer, EURASIP Meritorious Service Award, and National Science Foundation Young Investigator Award. He also received various teaching and research recognitions from the University of Maryland including a university-level Invention of the Year Award; and Poole and Kent Senior Faculty Teaching Award and Outstanding Faculty Research Award, both from A. James Clark School of Engineering Faculty. He is a Fellow of AAAS.