Forensic Detection of Image Manipulation Using Statistical Intrinsic Fingerprints

Matthew C. Stamm, *Student Member, IEEE*, and K. J. Ray Liu, *Fellow, IEEE*

***Abstract*—As the use of digital images has increased, so has the means and the incentive to create digital image forgeries. Accordingly, there is a great need for digital image forensic techniques capable of detecting image alterations and forged images. A number of image processing operations, such as histogram equalization or gamma correction, are equivalent to pixel value mappings. In this paper, we show that pixel value mappings leave behind statistical traces, which we shall refer to as a mapping's *intrinsic fingerprint*, in an image's pixel value histogram. We then propose forensic methods for detecting general forms globally and locally applied contrast enhancement as well as a method for identifying the use of histogram equalization by searching for the identifying features of each operation's intrinsic fingerprint. Additionally, we propose a method to detect the global addition of noise to a previously JPEG-compressed image by observing that the intrinsic fingerprint of a specific mapping will be altered if it is applied to an image's pixel values after the addition of noise. Through a number of simulations, we test the efficacy of each proposed forensic technique. Our simulation results show that aside from exceptional cases, all of our detection methods are able to correctly detect the use of their designated image processing operation with a probability of 99% given a false alarm probability of 7% or less.**

***Index Terms*—Contrast enhancement, digital forensics, digital image forgery, intrinsic fingerprints, pixel value histograms.**

I. Introduction

In recent years, digital images have become increasingly prevalent throughout society. Many governmental, legal, scientific, and news media organizations rely on digital images to make critical decisions or to use as photographic evidence of specific events. This proves to be problematic, as the rise of digital images has coincided with the widespread availability of image editing software. At present, an image forger can easily alter a digital image in a visually realistic manner. To avoid both embarrassment and legal ramifications, many of these organizations now desire some means of identifying image alterations and verifying image authenticity. As a result, the field of digital image forensics has been born.

One of the primary goals of digital image forensics is the identification of images and image regions which have undergone some form of manipulation or alteration. Because of the ill-posed nature of this problem, no universal method of detecting image forgeries exists. Instead, a number of techniques have been proposed to identify image alterations under a variety of scenarios. While each of these methods possesses their own limitations, it has been posited that if a large set of forensic methods are developed, it will be difficult for a forger to create an image capable of fooling all image authentication techniques [1].

Previous image forensic work has dealt with the identification of computer generated objects within an image [2] as well as detecting lighting angle inconsistencies [3], [4]. Inconsistencies in chromatic aberration [5] as well as the absence of color filter array (CFA) interpolation-induced correlations [6] have been used to identify inauthentic regions of an image. Classifier-based approaches have been proposed which identify image forgeries using a variety of statistical features [7]–[9]. Though these techniques are capable of detecting that an image has undergone some form of manipulation, they are unable to determine how an image has been altered beyond the identification of manipulated image regions.

One set of digital forensic techniques aimed at detecting image tampering has grown out of research into imaging device identification. Forensic imaging device identification methods attempt to determine the type of device used to capture an image, ascertain the device manufacturer or model, and identify the particular imaging device used [10]. These methods generally perform identification by estimating some device specific parameter such as CFA interpolation coefficients or sensor noise. Image forgery detection techniques have been proposed which operate by locating inconsistencies in these parameters [1], [11], or by using these parameters to estimate a tampering filter [12]. While these techniques are quite effective, they too suffer the drawback of being unable to identify the use of specific image altering operations.

It is important to note that most image altering operations leave behind distinct, traceable "fingerprints" in the form of image alteration artifacts. Because these fingerprints are often unique to each operation, an individual test to catch each type of image manipulation must be designed. While detecting image forgeries using these techniques requires performing a large set of operation-specific tests, these methods are able to provide insight into the specific operations used to manipulate an image. Prior work which identifies image tampering by detecting operation specific fingerprints includes the detection of resampling [13], double JPEG compression [14]–[16], as well as the parameterization of gamma correction [17]. Methods for detecting image forgeries have been proposed by detecting local abnormalities in an image's signal-to-noise ratio (SNR) [14]. Additionally, the efficient identification of copy and move forgeries has been studied [18].

Manuscript received October 11, 2009; revised May 21, 2010; accepted May 27, 2010. Date of publication June 17, 2010; date of current version August 13, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jessica J. Fidrich.

The authors are with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA (e-mail: mcstamm@umd.edu; kjrliu@umd.edu).

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.2010.2053202

In this work, we show that with the exception of the identity mapping, pixel value mappings leave behind statistical artifacts which are visible in an image's pixel value histogram. We refer to these artifacts as the *intrinsic fingerprint* of a pixel value mapping. By observing the common properties of the histograms of unaltered images, we are able to build a model of an unaltered image's pixel value histogram. We then use this model to identify diagnostic features of a pixel value mapping's intrinsic fingerprint. Because a number of image processing operations are in essence pixel value mappings, we propose a set of image forgery detection techniques which operate by detecting the intrinsic fingerprint of each operation. Specifically, we propose methods for detecting general forms globally and locally applied contrast enhancement, as well as a method for identifying the use of histogram equalization, a commonly used form of contrast enhancement. Additionally, we propose a method to detect the global addition of noise to a previously JPEG-compressed image by detailing the effect of noise on the fingerprint of a known pixel value mapping applied to the image in question.

While much of this work focuses on detecting operations which alter the perceptual qualities of an image as opposed to more obviously malicious tampering, detecting the image manipulations discussed in this work is still forensically significant. The detection of globally applied contrast enhancement provides insight into an image's processing history and may be useful prior information for other detection algorithms. Furthermore, contrast enhancement operations may be locally applied to disguise visual clues of image tampering. Localized detection of these operations can be used as evidence of cut-and-paste type forgery. Additive noise may be globally applied to an image not only to cover visual evidence of forgery, but also in an attempt to destroy forensically significant indicators of other tampering operations. Though the detection of these types of operations may not necessarily pertain to malicious tampering, they certainly throw in doubt the authenticity of the image and its content.

This paper is organized as follows. In Section II, we describe the forensically significant qualities of an unaltered image's pixel value histogram. In Section III, we define the intrinsic fingerprint of a pixel value mapping. We describe our proposed contrast enhancement detection techniques in Section IV. Included are methods for detecting both globally and locally applied contrast enhancement as well as a method for identifying histogram equalization. We develop a method for detecting the addition of noise to a previously JPEG-compressed image in Section V. Experiments designed to test the efficacy of each forensic scheme as well as simulation results are discussed after each detection method is proposed. We conclude this paper in Section VI.

II. System Model and Assumptions

In this work, we consider digital images created by using an electronic imaging device to capture a real world scene. We adopt the following model of the digital capture process. Each pixel is assigned a value by measuring the light intensity reflected from a real world scene onto an electronic sensor over the area pertaining to that pixel. Inherent in this process is the addition of some zero mean sensor noise which arises due to several phenomena including shot noise, dark current, and on-chip amplifier noise [19]. For color images, it is often the case that the light passes through a CFA so that only one color component is measured at each pixel location in this fashion. If this is the case, the color components not observed at each pixel are determined through interpolation. At the end of this process, the pixel values are quantized, then stored as the unaltered image. When analyzing a digital image, a histogram $h(l)$ of the color or gray level values l recorded at each pixel can be generated by creating L equally spaced bins which span the range of possible pixel values, then tabulating the number of pixels whose value falls within the range of each bin. Unless otherwise specified, we will hereafter assume that all gray level values lie in the set $\mathcal{P} = \{0, \ldots, 255\}$, all color values lie in the set $\mathcal{P}^3$, and that all pixel value histograms are calculated using 256 bins so that each bin corresponds to a unique gray or color layer value. After viewing the pixel value histograms of several camera generated images corresponding to a variety of scenes, we have observed that these histograms share common properties. None of the histograms contain sudden zeros or impulsive peaks. Furthermore, individual histogram values do not differ greatly from the histogram's envelope. To unify these properties, which arise due to observational noise [19], sampling effects, and complex lighting environments, we describe pixel value histograms as *interpolatably connected*. We denote an interpolatably connected histogram as one where any histogram value $h(l)$ can be approximated by $\hat{h}(l)$, the interpolated value of the histogram at pixel value l calculated using a cubic spline given $h(t)$ for all $t \in \mathcal{P} \backslash l$. The histogram of a typical unaltered image as well as its approximation $\hat{h}$, where each value of $\hat{h}$ has been calculated by removing a particular value from h then interpolating this value using a cubic spline, are shown in Fig. 1. As can be seen in this example, there is very little difference between the image's histogram and its approximation.

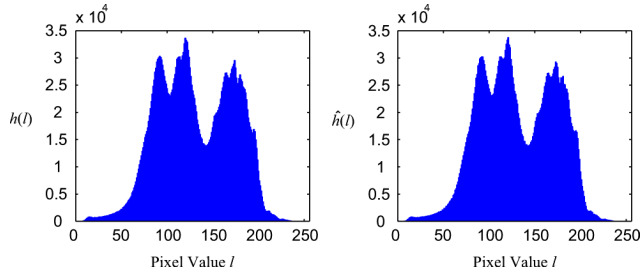

Fig. 1. Left: Histogram of a typical image. Right: Approximation of the histogram at left by sequentially removing then interpolating the value of each histogram entry.

To justify this model, we compiled a database of 341 unaltered images captured using a variety of digital cameras. We obtained each image's pixel value histogram h, as well as its approximated histogram $\hat{h}$, where each value $\hat{h}(x)$ was interpolated using cubic spline interpolation. We then calculated the mean squared error between $\hat{h}$ and h along with the signal power of h to obtain an SNR. The mean SNR of all image's histograms in the test database was 30.67 dB, reinforcing the notion that an image's pixel value histogram can be modeled as an interpolatably connected function.

There does exist one naturally occurring phenomena, which we refer to as *histogram saturation*, that may cause an unaltered

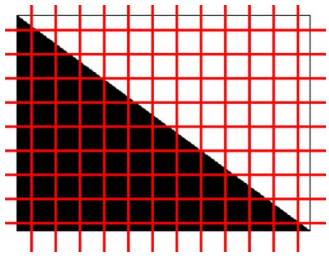

Fig. 2. Image sampling effects example.

image's pixel value histogram to contain an impulsive peak at one of two possible locations. High end histogram saturation effects occur in images corresponding to especially bright scenes where the dynamic range of the observed light intensity values extends well above the cutoff for the maximum pixel value. Because these pixels must be assigned the maximum pixel value of 255, a disproportionate number of pixels will take this value resulting in an impulsive peak at the high end of an image's histogram. Low end saturation effects occur in unusually dark images, where a large number of pixels taking the value 0 will cause an impulsive peak to occur at the low end of an image's histogram. While low end histogram saturation occurs less frequently than high end saturation, we have observed it in several unaltered images.

To explain why our histogram model is appropriate for digital images, consider the simple case of imaging a scene consisting of two distinct color regions shown in Fig. 2. Instinctively, we might assume that the histogram of this image would consist of zeros everywhere except for two impulses located at the pixel values corresponding to each of the two colors present in this scene. Such a histogram would obviously violate our model. In this scenario, however, the border between the color regions does not align with the pixel boundaries on the sensor of the imaging device, denoted by the grid. Many pixels lying along the color border correspond to sensor areas containing both colors. The resulting values of each of these pixels will lie in the convex hull of the values corresponding to each of the two colors present in the scene. The introduction of these new pixel values will effectively "smooth out" the pixel value histogram. Additionally, in the case of a color image, color values not observed at a particular pixel location must be interpolated because of the use of a CFA. The value of these interpolated pixels will also lie in the convex hull of their neighbors values and further smooth the histogram, resulting in one which is interpolatably connected.

Due to the complexity of real world scenes, it is exceedingly unlikely that the all color borders in an image will align directly with the pixel borders on an imaging device's sensor. Because of this, the effect described above should be present in virtually all real world images. Furthermore, additional factors contribute to the "connectivity" of pixel value histograms of images captured by digital cameras. The complex nature of most natural and man-made lighting environments rarely result in a real world scene consisting of several distinct colors with no shading. Instead, a continuum of colors and illumination levels normally exist. Furthermore, the presence of observational noise will slightly change the value of several pixels during the image capture process, thus further smoothing the histogram and resulting in one which is interpolatably connected.

III. Statistical Intrinsic Fingerprints of Pixel Value Mappings

A number of image processing operations, such as contrast enhancement, either include or can be specified entirely by a pixel value mapping. As is the case with most image processing operations, pixel value mappings leave behind distinct, forensically significant artifacts. These artifacts, which we will refer to as the *intrinsic fingerprint* of a pixel value mapping m, manifest themselves primarily in an image's pixel value histogram. To understand the effect of a pixel value mapping on an image's histogram, let us define $x \in \mathcal{P}$ as a pixel value present in an unaltered image and $y \in \mathcal{P}$ as the value that m maps x to such that

$$y = m(x). \tag{1}$$

Using this equation, the relationship between the pixel value histogram h_X of the unaltered image and the pixel value histogram h_Y of the same image after its pixel values have been subjected to the mapping m can be written as

$$h_Y(l) = \sum_{t=0}^{255} h_X(t)\mathbb{1}(m(t) = l) \tag{2}$$

where $\mathbb{1}(\cdot)$ denotes the indicator function. As a consequence, all entries in h_Y must take a value of either zero or the sum of several entries in h_X. Furthermore, any time n unaltered pixel values are mapped to the same output value, $n-1$ entries in h_Y must take a value of zero.

We now define the intrinsic fingerprint of m as

$$\begin{aligned} f_m(l) &= h_Y(l) - h_X(l) \\ &= \sum_{t=0}^{255} h_X(t)\mathbb{1}(m(t) = l) - h_X(l) \end{aligned} \tag{3}$$

which represents the change in the image's pixel value histogram due to the application of the mapping m. We can see that though the pixel value mapping is deterministic, its fingerprint depends on the image's histogram statistics. In subsequent sections it will be useful to examine a frequency domain representation of $f_m(l)$. Letting $\hat{m}(l) = m(l) - l$, the discrete Fourier transform (DFT) of $f_m(l)$ can be written as

$$\begin{aligned} F_m(k) &= \mathrm{DFT}\{f_m(l)\} \\ &= \sum_{l=0}^{255} h_X(l)\left(e^{-j\pi k\hat{m}(l)/128} - 1\right)e^{-j\pi kl/128} \\ &= -2j\sum_{l=0}^{255} h_X(l)\sin\left(k\frac{\pi\hat{m}(l)}{256}\right)e^{-j\pi k/128}\left(\frac{\hat{m}(l)}{2} + l\right). \end{aligned} \tag{4}$$

By examining (3) and (4), we can see that the intrinsic fingerprint is characterized not only by $m(l)$, but by $h_X(l)$ as well. Despite this, the intrinsic fingerprints left in two images with

different pixel value histograms will be quite similar. In the frequency domain, a mapping's tampering fingerprint consists of a linear combination of sinusoids whose frequencies are determined by $\hat{m}(l)$, which is nonzero only when $m(l) \neq l$. While the value of $h_X(l)$ affects the weight of each sinusoid in the summation, the presence and frequency of each sinusoid, and hence the basic structure of the intrinsic fingerprint, is determined by m.

Fig. 3 shows an example illustrating the similarity between fingerprints left in images with different pixel value histograms by a common mapping. As a reference, the pixel value histograms of a synthesized image with a uniform pixel values distribution and a typical image captured by a digital camera are shown in Figs. 3(a) and (b), respectively. The DFT of both histograms are shown in Figs. 3(c) and (d). As can be seen, these histograms differ significantly in both the pixel value and frequency domains. Next, the intrinsic fingerprints left in each image's histogram by the mapping

$$m(l) = \begin{cases} l, & \text{if } l \neq 100 \\ l+1, & \text{if } l = 100 \end{cases} \tag{5}$$

are compared. This mapping alters only one pixel value and is one of the simplest possible pixel value mappings. In this case, the intrinsic fingerprint left in each image's histogram will differ only by a scaling factor. This can be seen in Figs. 3(e) and (f), which show the magnitude of the frequency domain representation of each fingerprint. Finally, the intrinsic fingerprints left by the pixel value mapping

$$m(l) = \text{round}\left(\frac{7}{11}l\right) \tag{6}$$

are compared. This mapping is more complex than the previously considered mapping, and affects several pixel values. Figs. 3(g) and (f) show the magnitude of the frequency domain representation of each fingerprint. Though these fingerprints differ by more than a simple scaling factor, they share several identifying features including local peaks at $\omega = \pm 0.9081$, ± 1.8162, and ± 2.7243 radians, where

$$\omega = \begin{cases} \frac{k\pi}{128}, & \text{if } 0 \leq k < 128 \\ \frac{(k-256)\pi}{128}, & \text{if } 128 \leq k \leq 255. \end{cases} \tag{7}$$

In subsequent sections, we use intrinsic fingerprints along with our histogram model to identify evidence of image tampering. When examining a potentially altered image, if the histogram of unaltered pixel values is known, the tampering fingerprint can be obtained using (3). If the tampering fingerprint is zero for all l, one can conclude that a pixel value mapping was not used to alter the image. Alternatively, if the tampering fingerprint is nonzero for any values of l, it can be used to help determine the mapping used to alter the image. In most real scenarios, however, one has no *a priori* knowledge of an image's pixel value histogram, thus the tampering fingerprint cannot be calculated. Despite this, we are able to ascertain the presence of a tampering fingerprint by determining identifying features of a mapping's intrinsic fingerprint and searching for their presence in the histogram of the image in question. Furthermore, we reduce the number of false detections by using our histogram model to separate naturally occurring histogram features from those which correspond to a pixel value mapping's intrinsic fingerprint.

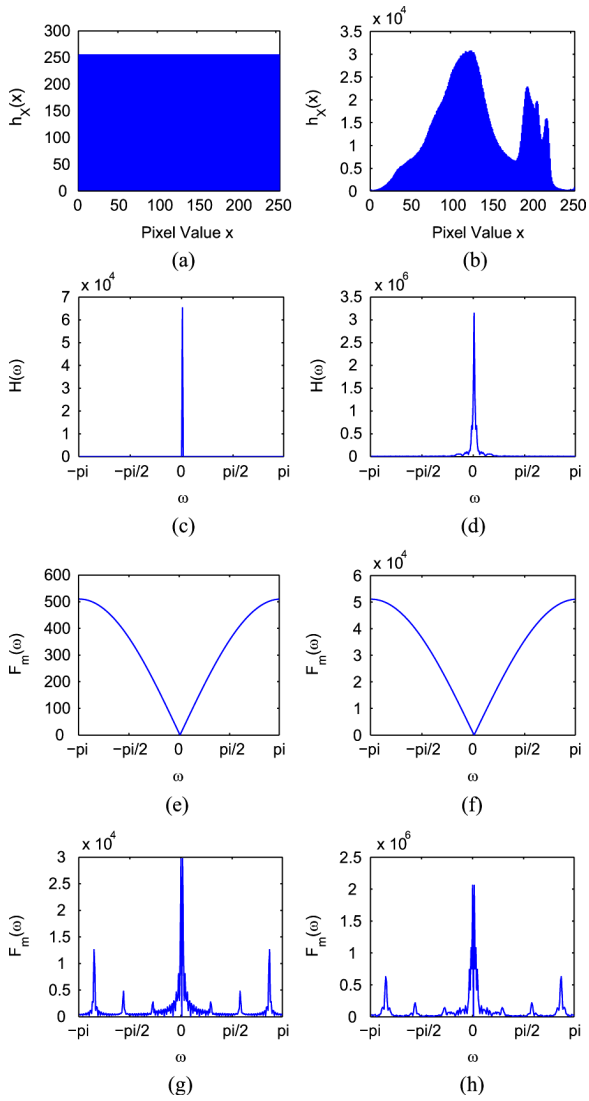

Fig. 3. Tampering fingerprint example showing the pixel value histograms of (a) a synthesized image with a uniform pixel distribution and (b) a real world image captured with a digital camera, the magnitude of the DFT of the histogram of (c) the synthesized image and (d) the real image, the magnitude of the frequency domain tampering fingerprints of (5) left in (e) the synthesized image and (f) the real image, as well as the magnitude of the frequency domain intrinsic fingerprints of (6) left in (e) the synthesized image and (f) the real image.

It is important to note that the concept of an intrinsic fingerprint extends to any monotonically increasing mapping, aside from the identity mapping, applied to discrete-valued data. For example, when an image undergoes double JPEG compression, its DCT coefficients are doubly quantized according to the mapping $y = q_2 \text{round}((q_1/q_2)\text{round}(x/q_1))$, where q_1 and q_2 are the quantization steps used. The periodic DCT coefficient histogram artifacts used in [14], [15], and [16] to identify double JPEG compression correspond to key features of the intrinsic fingerprint of this mapping. In fact, any time that an identifying feature of a mapping's intrinsic fingerprint can be determined, it can be used to detect the application of that mapping.

IV. Detecting Contrast Enhancement

In this section, we identify the intrinsic fingerprints of contrast enhancement mappings and use them to develop a set of image forensic techniques capable of detecting if an image has undergone contrast enhancement. While prior image forensic work has studied gamma correction [14], [17], this work assumes that the forensic examiner knows which specific type of contrast enhancement may have been applied and that the contrast enhancement mapping can be described by a simple parametric equation. Here, we present a detection approach which can be used to detect more general contrast enhancement operations and which requires no *a priori* knowledge of the form of contrast enhancement potentially applied. We begin by discussing a method for detecting the global application of contrast enhancement which operates by identifying histogram features indicative of general contrast enhancement fingerprints [20]. Next, we extend this technique into one capable of detecting locally applied contrast enhancement and show how it can be used to detect certain cut-and-paste image forgeries. Additionally we present a method for identifying the use of histogram equalization, a specific form of contrast enhancement, by identifying histogram features unique to its intrinsic fingerprint.

A. Detection of Globally Applied Contrast Enhancement

Contrast enhancement operations seek to increase the dynamic range of pixel values within an image. Most globally applied contrast enhancement operations accomplish this by applying a nonlinear mapping to the values of each pixel in the image, as described in Section III. In order to detect these operations, we must therefore detect the use of any pixel value mapping employed by a contrast enhancement operation. Without excluding any commonly used forms of contrast enhancement, we assume that all pixel value mappings in question are monotonically increasing. By considering only monotonic pixel value mappings, we purposefully exclude mappings which consist of a simple reordering of pixel values. As was previously mentioned, we detect the use of global contrast enhancement by identifying a characteristic feature of all monotonically increasing pixel value mappings (excluding the identity mapping), then use this feature in conjunction with our histogram model to ascertain whether the pixel value histogram of an image corresponds to a contrast enhanced image or an unaltered one.

In order to identify a diagnostic feature for contrast enhancement operations, let us first consider the effect of applying the mapping $m_{\tau+}$, defined as

$$m_{\tau+}(l) = \begin{cases} l, & \text{if } l \neq \tau \\ l+1, & \text{if } l = \tau \end{cases} \tag{8}$$

to an image with pixel value histogram h_X, resulting in an altered image with pixel value histogram h_Y. This mapping is significant because any monotonically increasing pixel value mapping, aside from the identity mapping, can be formed by the proper composition of the mappings $m_{\tau+}$ and $m_{\tau-}$ using various values of τ, where $m_{\tau-}$ is defined as

$$m_{\tau-}(l) = \begin{cases} l, & \text{if } l \neq \tau \\ l-1 & \text{if } l, = \tau. \end{cases} \tag{9}$$

Furthermore, let $h_X(\tau) = a$ and $h_X(\tau+1) = b$; therefore, after the mapping $m_{\tau+}$ is applied to the image, the altered image's histogram values at τ and $\tau+1$ will be $h_Y(\tau) = 0$ and $h_Y(\tau+1) = a+b$. The square of the Euclidean norm of h_X, denoted by $\|h_X\|_2^2$, will be less than that of h_Y because

$$\begin{aligned} \|h_X\|_2^2 &= \sum_l h_X(l)^2 \\ &= \sum_{l \neq \tau, \tau+1} h_X(l)^2 + a^2 + b^2 \\ &\leq \sum_{l \neq \tau, \tau+1} h_X(l)^2 + (a+b)^2 \\ &= \|h_Y\|_2^2. \end{aligned} \tag{10}$$

By Parseval's theorem, the energy of the DFT of h_Y must be greater than or equal to the energy of the DFT of h_X; however, this increase in energy cannot be realized in the DC coefficient because the total number of pixels in the image remains constant. An identical result can be proved for the mapping $m_{\tau-}$.

Because all monotonically increasing contrast enhancement mappings can be formed using the proper composition of the mappings $m_{\tau+}$ and $m_{\tau-}$, all contrast enhancement mappings result in an increase in energy within the image's pixel value histogram. This increase in energy corresponds to the energy of the intrinsic fingerprint left by the contrast enhancement mapping. In our experiments, we have observed that the increase in energy tends to be spread across the frequency spectrum, excluding the DC component which must remain constant. By contrast, since we model an unaltered image's histogram as an interpolatably connected function, we expect the histogram's DFT $H(k)$ to be a strongly low-pass signal. As a result, the presence of an appreciable amount of energy in the high frequency regions of $H(k)$ is indicative of contrast enhancement.

An alternate way of viewing this phenomena is to observe that locally contractive regions of a contrast enhancement mapping will cause multiple distinct input pixel values to be mapped to the same output pixel value. This will result in an isolated peak in the histogram of the contrast image at the common output pixel value. Similarly, locally expansive regions of a contrast enhancement mapping will cause adjacent input pixel values to be mapped apart, resulting in sudden gaps in the histogram of the enhanced image. Because these peaks and gaps are impulsive in nature, they will result in the presence of a significant high frequency component in $H(k)$. The bottom two plots of Fig. 4 show the frequency domain representations of the histogram of a typical image before and after it has undergone contrast enhancement.

Though an image's pixel value histogram is typically low-pass, this is not the case for an image whose histograms exhibit saturation effects. The impulsive component present in a saturated image's pixel value histogram will cause a DC offset to occur in its histogram's frequency domain representation which may be mistaken for the fingerprint of a contrast enhancement mapping. An example of this can be seen in Fig. 5, which shows a high end saturated image, its pixel value histogram, and the frequency domain representation of its histogram.

In light of these observations, we propose a technique which detects contrast enhancement by measuring the strength of

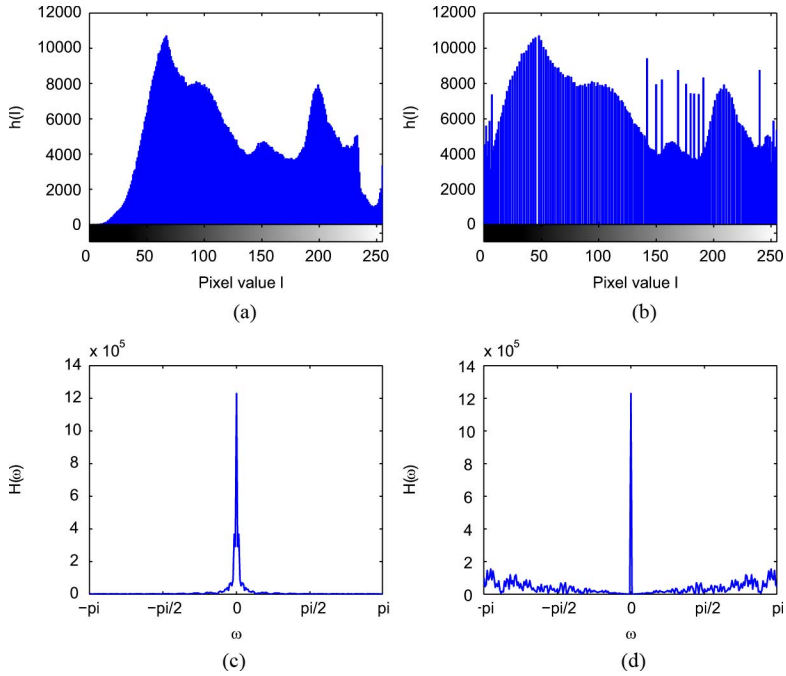

Fig. 4. Pixel value histogram of (a) an unaltered image and (b) the same image after contrast enhancement has been performed, as well as the magnitude of the DFT of (c) the unaltered image's histogram and (d) the contrast enhanced image's histogram.

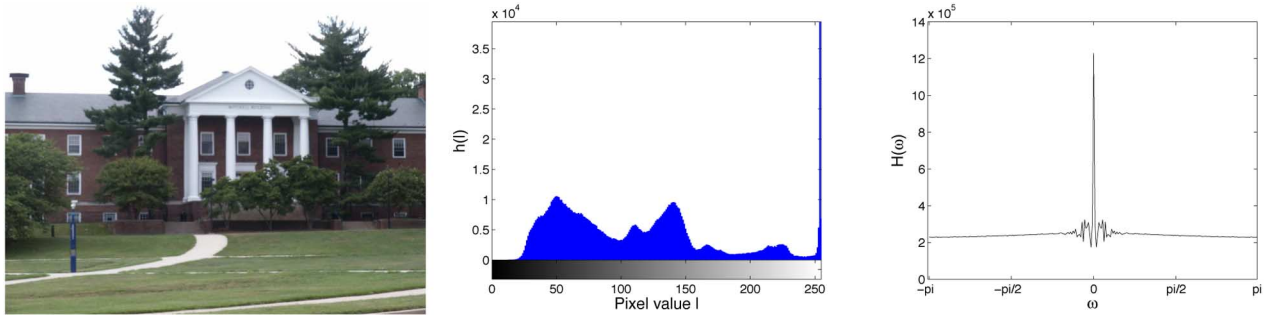

Fig. 5. Left: Image exhibiting high-end histogram saturation. Middle: Histogram of the image's green pixel values. Right: Magnitude of the DFT of the image's green pixel value histogram.

the high frequency components of an image's pixel value histogram, then comparing this measurement to a predefined threshold. To prevent unaltered images exhibiting histogram saturation effects from yielding large high frequency measurements indicative of contrast enhancement mapping fingerprints, we modify an image's histogram before testing so that it is free from saturation effects. This modified histogram $g(l)$ is obtained by performing the elementwise multiplication between $h(l)$ and a "'pinch off" function $p(l)$ so that

$$g(l) = p(l)h(l) \tag{11}$$

where

$$p(l) = \begin{cases} \frac{1}{2} - \frac{1}{2}\cos\left(\frac{\pi l}{N_p}\right), & l \leq N_p \\ \frac{1}{2} + \frac{1}{2}\cos\left(\frac{\pi(l-255+N_p)}{N_p}\right), & l \geq 255 - N_p \\ 1, & \text{else} \end{cases} \tag{12}$$

and N_p is the width of the region over which $p(l)$ decays from 1 to 0. The pinch off function is designed to both remove impulsive histogram components which may occur due to saturation effects as well as to minimize the frequency domain effects of multiplying $h(l)$ by $p(l)$, which behaves similar to a windowing function.

We calculate E, a normalized measure of the energy in the high frequency components of the pixel value histogram, from $g(l)$ according to the formula

$$E = \frac{1}{N}\sum_{k} |\beta(k)G(k)| \tag{13}$$

where N is the total number of pixels in the image, $G(k)$ is the DFT of $g(l)$, and $\beta(l)$ is a weighting function which takes values between 0 and 1. The purpose of $\beta(l)$ is to deemphasize low frequency regions of $G(l)$ where nonzero values do not necessarily correspond to contrast enhancement artifacts. In this work, we use the simple cutoff function

$$\beta(k) = \begin{cases} 1, & c \leq k \leq 128 \\ 0, & \text{else} \end{cases} \tag{14}$$

where c is the entry of the 256 point DFT corresponding to a desired cutoff frequency. $\beta(k)$ is zero for all values greater than

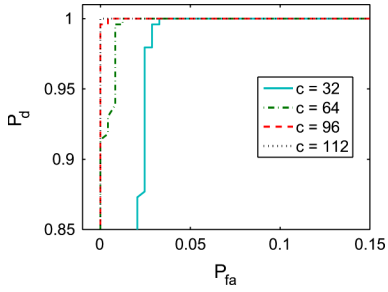

Fig. 6. Contrast enhancement detection ROC curves for images altered by a power law transformation with (b) $\gamma = 1.1$ using several values of the cutoff parameter c.

$k = 128$ because symmetry properties inherent in the DFT of real valued signals make it unnecessary to measure these values.

After F has been calculated, the decision rule δ_{ce} is used to classify an image as unaltered or contrast enhanced, such that

$$\delta_{\text{ce}} = \begin{cases} \text{image is } \textit{not} \text{ contrast enhanced}, & E < \eta_{\text{ce}} \\ \text{image is contrast enhanced}, & E \geq \eta_{\text{ce}}. \end{cases} \quad (15)$$

Our observation that an unaltered image's pixel value histogram is a strongly low-pass signal suggests that our detector's performance should improve as the frequency cutoff of c is increased. To verify this, we conducted an experiment on one set of data in which we obtained performance results for our contrast enhancement detection technique using c values ranging from 32 to 112 and compared the results. For this experiment, we used the green color layer from each of the 244 images in the Uncompressed Colour Image Database as a set of unaltered grayscale images [21]. We created a set of contrast enhanced images by applying the power law transformation

$$m(l) = 255\left(\frac{l}{255}\right)^{\gamma} \quad (16)$$

with $\gamma = 1.1$ to the pixel values of each of the unaltered images. We then classified each of these images as altered or unaltered using a series of decision thresholds and with the parameter $N_p = 4$. The probabilities of detection P_{d} and false alarm P_{fa} were determined for each threshold by respectively calculating the percent of contrast enhanced images correctly classified and the percent of unaltered images incorrectly classified. The series of receiver operating characteristic (ROC) curves displayed in Fig. 6 was generated using these results. As we hypothesized, our detection algorithm's performance improved as the value of c was increased, with the best performance being achieved when using $c = 112$.

To perform a larger scale test of our contrast enhancement detection technique, we compiled a database of 341 unaltered images consisting of many different subjects and captured under a variety of light conditions. These images were taken with several different cameras and range in size from 1500×1000 pixels to 2592×1944 pixels. The green color layer of each of these images was used to create a set of unaltered grayscale images. We applied the power law transformation defined in (16) to each of these unaltered grayscale images using γ values ranging from 0.5 to 2.0 to create a set of contrast enhanced images. Additionally, we modified each unaltered grayscale image using the nonstandard contrast enhancement mapping displayed in Fig. 7(a). These images were combined with the unaltered images to create a testing database of 4092 grayscale images.

To evaluate the performance of our contrast enhancement detection technique on this testing set, each image was classified as altered or unaltered using a series of decision thresholds. During classification, the parameters N_p and c were set to $N_p = 4$ and $c = 112$. As before, the detection and false alarm probabilities were calculated at each decision threshold and the series of ROC curves shown in Figs. 7(b) and (c) were generated. For each form of contrast enhancement tested, our detection technique acheived a P_{d} of 0.99 at a P_{fa} of approximately 0.03 or less.

B. Detection of Locally Applied Contrast Enhancement

Locally applied contrast enhancement can be defined as applying a contrast mapping to a set of contiguous pixels $\mathcal{J}$ within an image. If the cardinality of $\mathcal{J}$ is large enough that a histogram of the values of all pixels within $\mathcal{J}$ can be modeled as interpolatably connected, then when contrast enhancement is performed on the set $\mathcal{J}$ it will introduce its fingerprint into the histogram of $\mathcal{J}$. In light of this, the global contrast enhancement detection technique proposed in Section IV-A can be performed on a test set of pixels $\mathcal{J}'$ to achieve localized contrast enhancement detection.

Ideally, the test set $\mathcal{J}'$ should be identical to the set $\mathcal{J}$ when performing localized contrast enhancement detection. In reality, this is seldom the case because if an image contains a set of contrast enhanced pixels, the members of this set are not public knowledge. In some scenarios, the authenticity of a particular image region is thrown in doubt and the test set can be manually selected to correspond to encompass this region. If localized contrast enhancement is carefully applied, however, it will not be obvious which image regions have been altered and the image must be searched for contrast enhancement in its entirety. This can be performed by segmenting an image into a set of blocks so that each block corresponds to a unique test set, then performing contrast enhancement detection on each block. The blockwise detection results can be combined to identify image regions which show signs of contrast enhancement.

In some scenarios, locally applied contrast enhancement detection can be used to identify other, more obviously malicious image manipulations such as cut-and-paste forgery. Cut-and-paste image forgery consists of creating a composite image by replacing a contiguous set of pixels in one image with a set of pixels $\mathcal{O}$ corresponding to an object from a separate image. If the two images used to create the composite image were captured under different lighting environments, an image forger may need to perform contrast enhancement on O so that lighting conditions match across the composite image. Failure to do this may result in a composite image which does not appear realistic. Image forgeries created in this manner can be identified by using localized contrast enhancement detection to locate $\mathcal{O}$, the cut-and-pasted region.

When performing blockwise localized contrast enhancement detection, it is important to ensure that the testing blocks

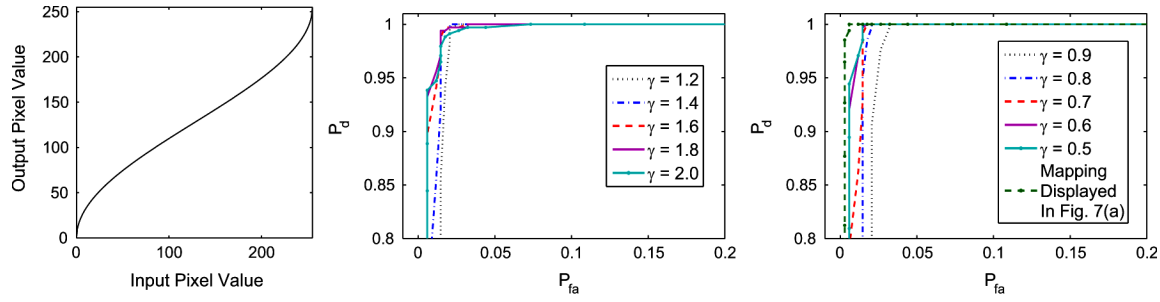

Fig. 7. Contrast enhancement detection ROC curves for images altered by a power law transformation with (b) $2.0 \geq \gamma \geq 1.2$, and (c) $0.5 \geq \gamma \geq 0.9$ as well as the mapping displayed in (a).

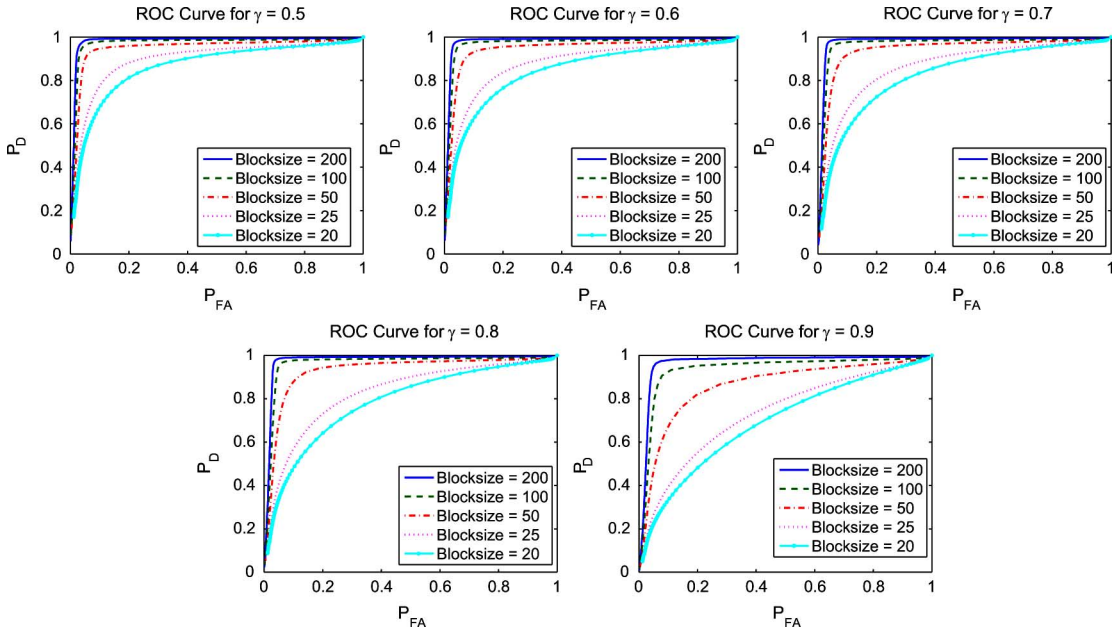

Fig. 8. ROC curves obtained using different testing block sizes for images altered by a power law transformation with $\gamma = 0.5$ (top left), $\gamma = 0.6$ (top middle), $\gamma = 0.7$ (top right), $\gamma = 0.8$ (bottom left), and $\gamma = 0.9$ (bottom right).

are large enough to yield histograms suitable for contrast enhancement detection. If the blocks are too small, they may not contain enough pixels for the interpolatably connected histogram model to hold valid. In order to determine which block sizes are sufficient to perform reliable detection and examine the effectiveness of the local contrast enhancement detection scheme, we performed the following experiment. Each of the 341 unaltered images from the second test database described in Section IV-A along with the power law transformed images corresponding to $\gamma = 0.5$ through 0.9 were segmented into square blocks. This process was performed for blocks of size 200×200, 100×100, 50×50, 25×25, and 20×20 pixels. Each block was then classified as contrast enhanced or unaltered by our contrast enhancement detection scheme using a variety of different thresholds. False alarm and detection probabilities were determined at each threshold and for every choice of block size by calculating the percent of incorrectly classified unaltered blocks and the percent of correctly classified contrast enhanced blocks respectively. This information was used to generate a set of ROC curves, shown in Fig. 8 for each value of γ which was tested.

The ROC curves shown in Fig. 8 indicate that local contrast enhancement can be reliably detected using testing blocks sized at least 100×100 pixels. At a P_{fa} of approximately 5%, a P_{d} of at least 95% was achieved using 200×200 pixel blocks and a P_{d} of at least 80% was achieved using 100×100 pixel blocks for each form of contrast enhancement tested. These results improved markedly when the contrast enhancement applied was stronger than the relatively mild power law transformation using $\gamma = 0.9$. In such cases, a P_{d} of roughly 98.5% and 96% was achieved with a P_{fa} of aproximatley 5% for blocks sized 200×200 pixels and 100×100 pixels, respectively. It should also be noted that testing blocks sized 25×25 pixels and smaller appear to contain an insufficient number of pixels to perform reliable contrast enhancement detection.

An example of a cut-and-paste image forgery in which the pasted region has undergone contrast enhancement is shown in Fig. 9 along with the localized contrast enhancement detection results obtained from our proposed forensic technique. Adobe Photoshop was used to create the forged image shown in Fig. 9(c) from the unaltered images shown in Figs. 9(a) and (b). In order to detect the forgery, the image was then segmented into

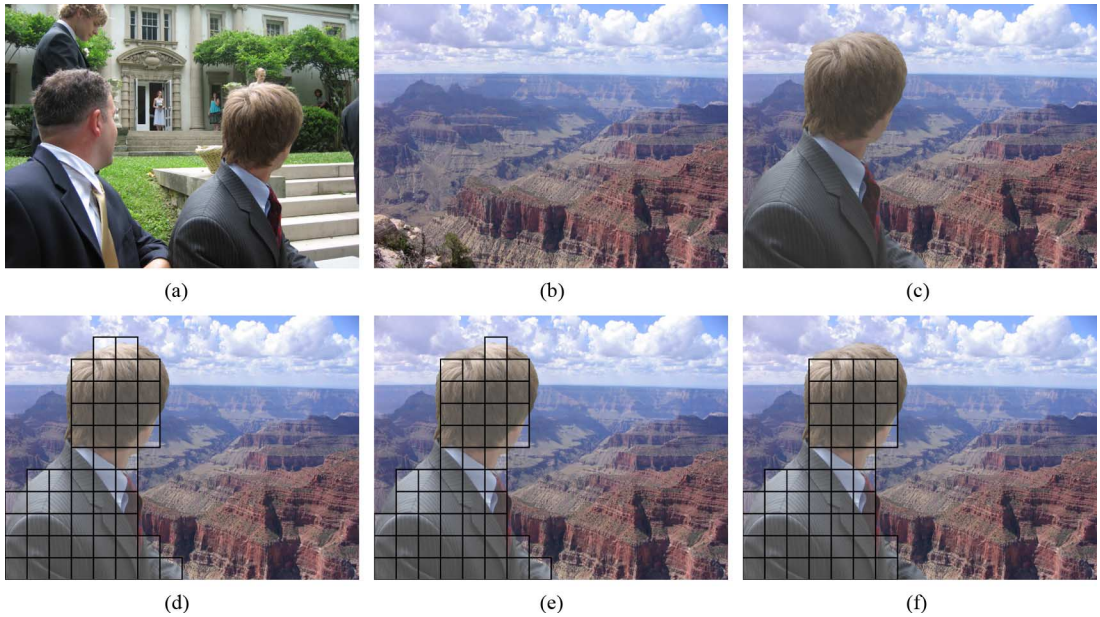

Fig. 9. Cut and paste forgery detection example showing (a) the unaltered image from which an object is cut, (b) the unaltered image into which the cut object is pasted, (c) the composite image, (d) red layer blockwise detections, (e) green layer blockwise detections, and (f) blue layer blockwise detections. Blocks detected as contrast enhanced are highlighted and boxed.

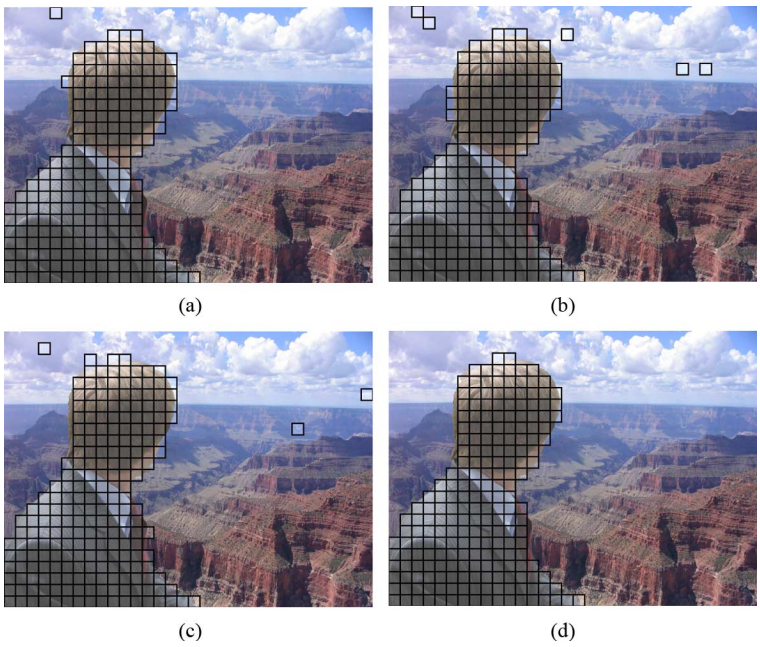

Fig. 10. Cut and paste forgery detection results using 50×50 pixel blocks showing (a) red layer blockwise detections, (b) green layer blockwise detections, (c) blue layer blockwise detections, and (d) blockwise detections that occur across all three color layers.

100×100 pixel blocks, each of which was tested for evidence of locally applied contrast enhancement. Figs. 9(d)–(f) show the results of performing localized contrast enhancement detection on the red, green, and blue color layers of the composite image. Blocks corresponding to contrast enhancement detections are highlighted and outlined in black. In this example, each of these blocks contain pixels that correspond to the inauthentic object.

Fig. 10 shows detection results when the block size is reduced to 50×50 pixels. When detection was performed separately on each color layer of the forged image, several false alarms occurred, as can be seen in Figs. 10(a)–(c). These false alarm blocks generally correspond to areas of the sky where the pixel values are nearly constant, leading to a pixel value histogram that contains an impulsive component outside of the pinch off region. The number of false alarms can be reduced in color images such as this by classifying a block as contrast enhanced only if a detection occurs at the corresponding block in each of the three color layers. Fig. 10(d) shows the results of applying this detection criteria to the single color layer detections displayed in in Figs. 10(a)–(c).

C. Histogram Equalization

In some scenarios, it may be desirable to identify the specific form contrast enhancement used to modify an image. One simple and commonly used form of contrast enhancement is histogram equalization. Histogram equalization effectively increases the dynamic range of an image's pixel values by subjecting them to a mapping such that the distribution of output pixel values is approximately uniform [22]. The mapping used to accomplish this is dependent upon the histogram of the unaltered image and is is generated according to the equation

$$m_{he}(l) = \text{round}\left(255\sum_{t=0}^{l}\frac{h(t)}{N}\right) \tag{17}$$

where N is the total number of pixels in the image. Because the histogram of an unaltered image does not normally approximate a uniform distribution, the "uniformity" of an equalized image's histogram can be used as an identifying feature of this mapping's intrinsic fingerprint. We propose a test which measures the distance between an image's normalized histogram and the uniform distribution, then uses this distance to determine if the image has undergone histogram equalization. This test can be used after contrast enhancement has been detected or it can performed independently of our generalized contrast enhancement detection technique.

Like any other contrast enhancement mapping, histogram equalization will introduce zeros into an image's pixel value histogram through the process discussed in Section III. Because of this, measures such as the Kullback–Leibler divergence are ill equipped to determine the distance between an image's normalized histogram of an equalized image and the uniform distribution. Similarly, other measures such as the mean absolute difference or the mean squared difference between an image's normalized histogram and the uniform distribution will be biased away from small values indicative of a uniform histogram by the zeros and accompanying impulsive peaks present in an equalized image's histogram. To mitigate this problem, we propose measuring the uniformity of an image's histogram in the frequency domain, where histogram equalization's identifying features can be separated from other obfuscating effects.

The frequency domain representation of a constant function is an impulse centered at zero. Using this fact, we obtain a frequency domain measure of the distance D of an image's normalized histogram from the uniform distribution according to the formula

$$D = \frac{1}{N}\left(\sum_{k\neq 0}|H(k)|\alpha(k)\right). \tag{18}$$

In (18), $\alpha(k)$ is a weighting function used to deemphasize the high frequency regions in $H(k)$ where the energy introduced by histogram equalizations intrinsic fingerprint tends to accumulate. After calculating D for an image in question, the decision rule δ_{he} is then used to determine if histogram equalization has been performed, as in (19), shown at the bottom of the page, where η_{he} is the decision threshold.

As discussed in Section IV-A, frequency domain detection methods suffer problems due to the constant offset present in $H(k)$ in high and low end histogram saturated images. Multiplying $h(l)$ by a pinch off function will not remove the effects of histogram saturation because for histogram equalized images, the location of the impulsive component is often shifted by histogram equalization. Instead, we identify impulsive components which are likely due to saturation effects and remove them to obtain a modified histogram.

For low end histogram saturated images, we may safely assume that before histogram equalization is applied to an image, the impulsive nature of its histogram will cause the number of pixels in the lowest bin to be greater than $2N/255$. After histogram equalization is performed, the pixel value $l = 0$ will be mapped to an output value greater than or equal to 2 because

$$\begin{aligned} m_{he}(0) &= \text{round}\left(255\sum_{t=0}^{l=0}\frac{h(t)}{N}\right) \\ &\geq \text{round}\left(255\left(\frac{2}{255}\right)\right) = 2. \end{aligned} \tag{20}$$

Letting l' denote the lowest value of l such that $h(l) > 0$, images which may be of this nature can be identified if $l' \geq 2$ and $h(l') \geq 2N/255$. For these images, the effects of the impulsive histogram component can be mitigated by forming a new histogram $h'(l)$ by retaining only the section of the histogram corresponding to pixel values larger than the kth nonzero entry. More explicitly, $h'(l)$ can be defined as $h'(l) = h(l'_k + l + 1)$, where l'_k is the kth nonempty bin in $h(l)$. The parameter $h(l)$ in (18) can then be replaced by $h'(l)$ to obtain a value of D unbiased by low end histogram saturations effects.

In the case of high end histogram saturated images, we can similarly assume that $h(255) \geq 2N/255$. When histogram equalization is performed on these images, the input pixel value $l = 254$ is mapped to an output value of 253 or less because

$$\begin{aligned} m_{he}(254) &= \text{round}\left(255\sum_{t=0}^{l=254}\frac{h(t)}{N}\right) \\ &\leq \text{round}\left(255\left(1-\frac{2}{255}\right)\right) = 253. \end{aligned} \tag{21}$$

Using this information, high end saturated images that may have undergone histogram equalization can be identified by determining if $l'' \leq 253$ and $h(255) \geq 2N/255$, where l'' is the

$$\delta_{he} = \begin{cases} \text{histogram equalization } \textit{not} \text{ present}, & D > \eta_{he} \\ \text{histogram equalization present}, & D \leq \eta_{he} \end{cases} \tag{19}$$

largest value of l such that $l'' < 255$ and $h(l'') > 0$. A new histogram that does not contain the impulsive histogram component can now be formed by letting $h''(l) = h(l)$ for $l = 0, \ldots, l''_k - 1$, where l''_k is the kth nonempty bin in $h(l)$ counting backwards from $l = 255$. As before, $h(l)$ in (18) can be replaced by $h''(l)$ to achieve a value of D unbiased by high end histogram saturations effects.

To evaluate the performance of our histogram equalization classification method, we performed histogram equalization on the 341 unaltered grayscale images from our global contrast enhancement test database described in Section IV-A. We combined the histogram equalized images with their unaltered counterparts to create a histogram equalization testing database. Next we used our detection algorithm to determine if each image in the database had undergone histogram equalization. Detection was performed using two different weighting functions

$$\alpha_1(k) = \begin{cases} \exp(-r_1 k), & \text{if } 0 \le k < 128 \\ \exp(-r_1(256-k)), & \text{if } 128 \le k \le 255 \end{cases} \quad (22)$$

with r_1 taking values between 0.1 and 0.5 and

$$\alpha_2(k) = \begin{cases} 1, & \text{if } k \le r_2 \text{ or } (256-k) \le r \\ 0, & \text{else} \end{cases} \quad (23)$$

with r_2 values ranging from 4 to 16. The false alarm and detection probabilities were then determined by calculating the percentage of incorrectly classified unaltered images and the percentage of correctly classified histogram equalized images respectively.

A series of ROC curves showing the performance of our histogram equalization detection scheme are displayed in Fig. 11. Our detector achieved its best performance using $\alpha_1(k)$ as a weighting function with $r_1 = 0.5$. Under these conditions, a $P_{\rm d}$ of 99% was reached with a $P_{\rm fa}$ of approximately 0.5% as well as a $P_{\rm d}$ of 100% with a $P_{\rm fa}$ of nearly 3%. Additionally, Fig. 11 shows that our detection scheme's performance improved as the value of r_1 increased when using $\alpha_1(k)$, and as the value of r_2 decreased when using $\alpha_2(k)$. Both of these trends correspond to an increase in detector performance as the weighting function is chosen to place more emphasis on low frequency regions of $H(k)$ during detection. This reinforces the notion that a weighting function is needed to deemphasize the middle and high frequency regions of $H(k)$ where general contrast enhancement artifacts can obscure evidence of histogram equalization.

Additionally, we performed an experiment to verify that our histogram equalization classification technique can differentiate between histogram equalization and other forms of contrast enhancement. We created a new testing database consisting of the 341 unaltered grayscale images as well as 1705 of the gamma corrected images corresponding to $\gamma = 0.5$ to 0.9 from the experiment discussed in Section IV-A. We then used our histogram equalization detection test to classify each of the images in the database as histogram equalized or not equalized. During classification, the weighting function described in (22) was used with $r_2 = 4$. The probabilities of detection and false alarm were obtained by calculating the percentage of correctly classified histogram equalized images and incorrectly classified gamma corrected images respectively. These probabilities were then used

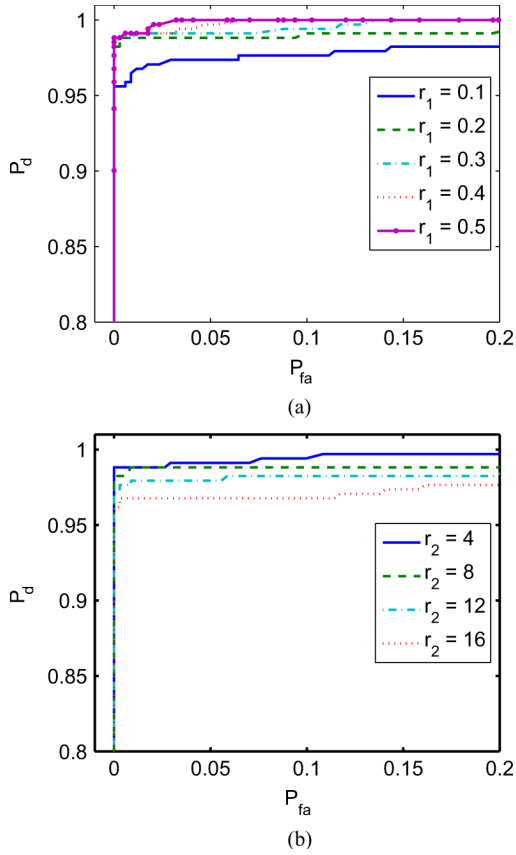

Fig. 11. Histogram equalization detection ROC curves obtained (a) using the weighting function defined in (22) and (b) using the weighting function defined in (23).

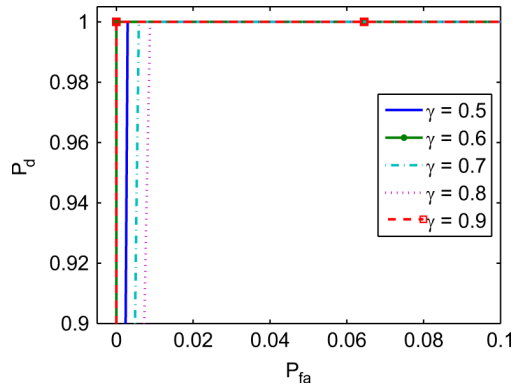

Fig. 12. ROC curves obtained when differentiating between histogram equalization and other forms of contrast enhancement.

to generate the ROC curves displayed in Fig. 12. A $P_{\rm d}$ of 100% was achieved at a $P_{\rm fa}$ of less than 1% for each form of contrast enhancement tested.

V. Detecting Additive Noise in Previously JPEG-Compressed Images

In this section, we present a technique designed to detect the global addition of noise to an image that has previously undergone JPEG compression. Though this may initially seem to be a fairly harmless operation, additive noise can be used to disguise visual traces of image forgery or in an attempt to mask statistical

artifacts left behind by other image altering operations. Previous work has dealt with the detection of noise added to specific regions of an image by searching for fluctuations in localized estimates of an image's SNR [14]. This method fails, however, when noise has been globally added to an image because this scenario will not result in localized SNR variations. Instead of relying upon SNR measurements, our proposed technique operates by applying a predefined mapping with a known fingerprint to a potentially altered image's pixel values [23]. This mapping is chosen such that an identifying feature of its fingerprint will be absent if noise was added to the image. Accordingly, we are able to detect the presence of additive noise if the application of the predefined mapping does not introduce a fingerprint with this feature.

A. Scale and Round Mapping

To perform additive noise detection, we make use of a mapping which we refer to as the *scale and round* mapping. We define the scale and round mapping as

$$v = \text{round}(cu) \tag{24}$$

where $u, v \in \mathbb{Z}$ and c is a fixed scalar. To understand the fingerprint left by this mapping, let us also define $\mathcal{U}_c(v)$ as the set of u values mapped to each distinct v value by (24), where

$$\mathcal{U}_c(v) = \{u | v = \text{round}(cu)\}. \tag{25}$$

The cardinality of this set, denoted by $|\mathcal{U}_c(v)|$, depends on the values of both c and v. It can be proven that if $c = p/q$ such that $p, q \in \mathbb{Z}$ are relatively prime, $|\mathcal{U}_c(v)|$ is periodic in v with period p. To see why this is so, consider first the following two easily proven lemmas:

Lemma 1: Given $a \in \mathbb{Z}$ and $b \in \mathbb{R}$

$$a = \text{round}(b) \Leftrightarrow a + k = \text{round}(b + k),\ \forall k \in \mathbb{Z}. \tag{26}$$

Lemma 2: Given u, $v \in \mathbb{Z}$, and $c = p/q$ such that $p, q \in \mathbb{Z}$ are relatively prime

$$v = \text{round}(cu) \Leftrightarrow v + p = \text{round}(c(u + q)). \tag{27}$$

Now using Lemma 2, we can state that for all $u \in \mathcal{U}_c(v)$, there exists some $\tilde{u} \in \mathcal{U}_c(v + p)$, namely $\tilde{u} = u + q$, which implies that $|\mathcal{U}_c(v)| = |\mathcal{U}_c(v+p)|$. This proves that the number of u values mapped to each v value is periodic with period p. As a consequence, the intrinsic fingerprint of the scale and round operation will contain a periodic component with period p.

B. Hypothesis Testing Scenario

We now shift our discussion to JPEG compression and its significance to the detection of additive noise. When a color image undergoes JPEG compression, each pixel in the image is first converted from the RGB color space to the YCbCr color space using a linear transformation. Next, each color layer is divided into a series of 8×8 pixel blocks and the discrete cosine transform of each block is computed. The resulting set of DCT coefficients are quantized by dividing each coefficient by its corresponding entry in a quantization matrix, then rounding the result to the nearest integer. Finally, the quantized DCT coefficients are reordered into a single bitstream which is losslessly compressed.

The image is decompressed by losslessly decoding the bitstream of quantized DCT coefficients, then reshaping it back into the series of blocks. The DCT coefficients are dequantized by multiplying each quantized DCT coefficient by its corresponding entry in the quantization matrix used during compression. Next, the inverse DCT (IDCT) of each block is computed, resulting in a set of pixel values in the YCbCr color space. Because the dequantized DCT coefficients are integer multiples of their respective quantization table entries and because the IDCT is a fixed linear transformation, the pixel values in the YCbCr color space will lie in a countable subset of $\mathbb{R}^3$. As a result, if a monotonically increasing mapping is applied to any color layer in the YCbCr color space, that mapping's fingerprint will be introduced into the histogram of the color layer's values.

In the final stage of JPEG decompression, the pixels are transformed from the YCbCr to the RGB color space, then projected back into $\mathcal{P}^3$. Letting $\mathbf{y}$ denote a pixel in the RGB color space, $\mathbf{x}$ denote the same pixel in the YCbCr color space, and $\mathbf{T}$ be the linear transformation that maps a pixel from the YCbCr to the RGB color space, this process can be described mathematically by the equation

$$\mathbf{y} = \text{truncate}(\text{round}(\mathbf{Tx})) \tag{28}$$

where the operation $\text{truncate}(\cdot)$ maps values of its argument less than 0 to 0 and values greater than 255 to 255. By defining $Q(\mathbf{Tx}) = \text{truncate}(\text{round}(\mathbf{Tx})) - \mathbf{Tx}$, we may now formulate the detection of additive noise as the following hypothesis testing problem:

$$\begin{aligned} H_0 &: \mathbf{y} = \mathbf{Tx} + Q(\mathbf{Tx}) \\ H_1 &: \mathbf{y} = \mathbf{Tx} + Q(\mathbf{Tx}) + \mathbf{n}. \end{aligned} \tag{29}$$

It should be noted that traditional Bayesian techniques cannot be used to differentiate between these two hypotheses because the distribution of $\mathbf{x}$ is unknown. Instead, we differentiate between these two hypotheses by observing that the fingerprint left by the mapping

$$\mathbf{z} = \text{round}(c\mathbf{T}^{-1}\mathbf{y}) \tag{30}$$

where the constant $c = p/q$ is such that $p, q \in \mathbb{Z}$ are relatively prime, differs under each hypothesis. When this mapping is applied to each pixel within an image, the hypothesis testing problem outlined in (29) can be rewritten as

$$\begin{aligned} H_0 &: \mathbf{z} = \text{round}(c\mathbf{x} + \mathbf{e}) \\ H_1 &: \mathbf{z} = \text{round}(c\mathbf{x} + \mathbf{e} + c\mathbf{T}^{-1}\mathbf{n}) \end{aligned} \tag{31}$$

where $\mathbf{e} = c\mathbf{T}^{-1}Q(\mathbf{Tx})$.

Under hypothesis H_0, the ith entry of $\mathbf{z}$ can be expressed as according to the formula

$$\begin{aligned} z_i &= \text{round}(cx_i + e_i) \\ &= \text{round}(cx_i) + \text{round}(e_i) + d_i \end{aligned} \tag{32}$$

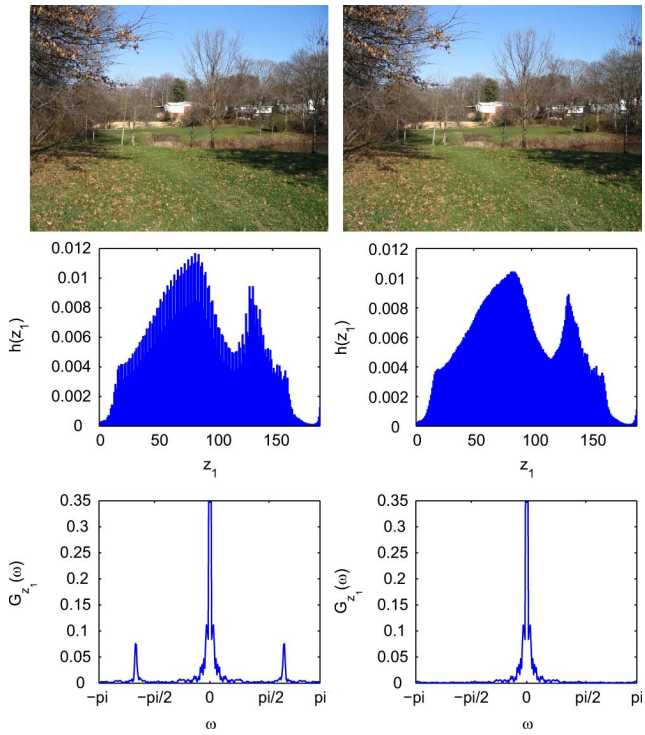

Fig. 13. Example showing an unaltered image (top left), its normalized z_1 histogram (middle left), and the magnitude of the DFT of its z_1 histogram (bottom left), as well as an altered version of the image to which unit variance Gaussian noise has been added (top right), its normalized z_1 histogram (middle right), and the magnitude of the DFT of its z_1 histogram (bottom right). In both cases, the scaling parameter was chosen to be $c = 3/4$.

where d_i is an independent random variable which accounts for the error induced by summing the individually rounded terms cx_i and e_i. Because the variances of the terms $\mathrm{round}(e_i)$ and d_i are typically small, the term $\mathrm{round}(cx_i)$ dominates the behavior of the PMF of z_i. Since the term $\mathrm{round}(cx_i)$ is of the same form as (24), the number of distinct x_i values mapped to each z_i value will occur in a fixed periodic pattern. This will result in the presence of a discernible periodic pattern with period p in the envelope of the histogram of z_i values. This pattern, which corresponds to the intrinsic fingerprint of the scale and round mapping, can be clearly seen in Fig. 13.

Under hypothesis H_1, we find that the histogram of z_i values exhibits different behavior. Defining the matrix $\mathbf{W}$ as the inverse of $\mathbf{T}$ such that

$$\mathbf{T}^{-1} = \mathbf{W} = \begin{bmatrix} W_{1,1} & W_{1,2} & W_{1,3} \\ W_{2,1} & W_{2,2} & W_{2,3} \\ W_{3,1} & W_{3,2} & W_{3,3} \end{bmatrix} \tag{33}$$

the ith entry of $\mathbf{z}$ can be expressed as

$$\begin{aligned} z_i &= \mathrm{round}\left(cx_i + \sum_{j=1}^{3} cW_{i,j}n_j + e_i \right) \\ &= \mathrm{round}(cx_i) + \sum_{j=1}^{3} \mathrm{round}\left(cW_{i,j}n_j\right) \\ &\quad + \mathrm{round}(e_i) + d_i \end{aligned} \tag{34}$$

where d_i is an independent random variable which accounts for the error induced by moving the summation of terms outside the round operation. Under this hypothesis, the PMF of z_i is equivalent to the convolution of each of these terms. Under this hypothesis, however, three additional terms containing the scale and round mapping appear, each with their own scaling constant $cW_{i,j}$. If these scaling constants along with the original scaling constant c are such that the fingerprints introduced into each individual term share no common period, then the convolution of the PMFs of each term will effectively smooth out the PMF of z_i. As a result, no periodic pattern will be introduced into the histogram of z_i by the mapping defined in (28). This effect can be observed in the example shown in Fig. 13.

C. Additive Noise Detection in Images

Using this information, we are able to rephrase the detection of the addition of noise to a previously JPEG-compressed image as the detection of the the periodic fingerprint of (28) within the envelope of $h_{z_i}(l)$, the normalized histogram of z_i. Because of its periodic nature, the detection of this fingerprint is particularly well suited for the frequency domain, where it will produce a peak centered at the frequency bin corresponding to its fundamental frequency or an integer multiple thereof. The bottom two plots of Fig. 13 show the presence or absence of this peak under each hypothesis. Furthermore, since the period of the fingerprint is dictated by our choice of the scaling constant, we are able to choose the frequency location of this peak.

To facilitate detection, we obtain a frequency domain representation $G_{z_i}(k)$ of the histogram $h_{z_i}(l)$ which is free from any possible high or low end histogram saturation effects. We accomplish this by defining $G_{z_i}(k)$ as the DFT of $g_{z_i}(l)$, which we calculate using the equation

$$g_{z_i}(l) = h_{z_i}(l)p(l) \tag{35}$$

where $p(l)$ is the pinch off function denoted in (12). Next, we test for the presence of the periodic fingerprint by measuring the strength of the peak that it introduces into $G_{z_i}(k)$. This measurement is obtained using the equation

$$S = \min\left\{ \frac{|G_{z_i}(k^*)|}{\frac{1}{|\mathcal{B}_1|}\sum_{j\in\mathcal{B}_1}|G_{z_i}(j)|}, \frac{|G_{z_i}(k^*)|}{\frac{1}{|\mathcal{B}_2|}\sum_{j\in\mathcal{B}_2}|G_{z_i}(j)|} \right\} \tag{36}$$

where k^* is the frequency location of the expected peak and $\mathcal{B}_1$ and $\mathcal{B}_2$ are sets of contiguous indices of G_{z_i} lying above and below k^* respectively. Finally, we use a decision rule δ_n corresponding to the threshold test

$$\delta_n = \begin{cases} \text{noise has } \textit{not} \text{ been added}, & \text{if } S < \eta_n \\ \text{noise has been added}, & \text{if } S \geq \eta_n \end{cases} \tag{37}$$

to determine the presence or absence of additive noise within the image.

When using this technique, the sets $\mathcal{B}_1$ and $\mathcal{B}_2$ should be chosen such that they do not include indices directly adjacent to k^*. This is because DFT windowing effects may result in artificially larger values of $|G_{z_i}(k)|$ around the peak if it is present. Additionally, the interpolatable connectivity restriction placed

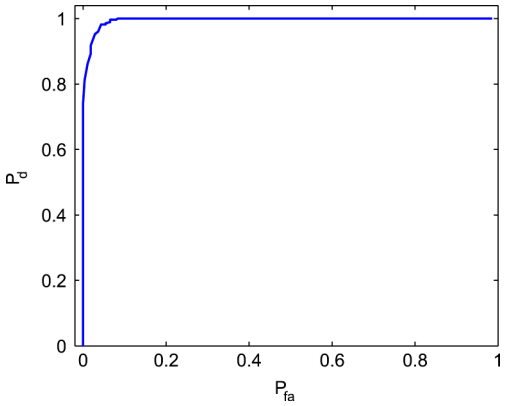

Fig. 14. Additive noise detection ROC curve for images which were JPEG-compressed using default camera settings then altered by adding unit variance Gaussian additive noise.

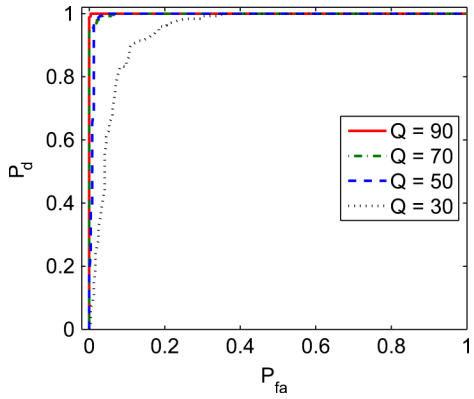

Fig. 15. Additive noise detection ROC curve for images which were JPEG-compressed at several different quality factors then altered by adding unit variance Gaussian additive noise.

upon the histogram of pixel values in our image model implies that $G_{z_i}(k)$ will be strongly low-pass in nature. This property suggests that to achieve better differentiability, c should be chosen such that it introduces a high frequency signal into $h_{z_i}(l)$.

To evaluate the performance of our additive noise detection technique, we compiled a set of 277 unaltered images taken by four different digital cameras from unique manufacturers. These images capture a variety of different scenes and were saved as JPEG-compressed images using each camera's default settings. A set of altered images was created by decompressing each image and independently adding unit variance Gaussian noise to each pixel value. These altered images were then saved as bitmaps, along with decompressed versions of the original images, creating a testing database of 554 images. Next we used our additive noise detection test to determine if noise had been added to each image in the database. When creating the histogram of z_i values, we chose $i = 1$ which corresponds to using the luminance or "Y" component of each pixel. The parameter c was chosen to take the value $c = 7/11$ leading to an expected peak location of $k^* = 71$. The sets of $\mathcal{B}_1$ and $\mathcal{B}_2$ were chosen to be $\mathcal{B}_1 = \{61, \ldots, 68\}$ and $\mathcal{B}_2 = \{74, \ldots, 81\}$.

Detection and false alarm probabilities were determined at a series of decision thresholds by calculating the percentages of correctly classified images to which noise had been added and incorrectly classified unaltered images, respectively. Using this data, an ROC curve showing the performance of our additive noise detection algorithm is displayed in Fig. 14. A $P_{\rm d}$ of approximately 80% was achieved at a false alarm rate less than 0.4%. When the $P_{\rm fa}$ was held less than 6.5%, the $P_{\rm d}$ increased to nearly 99%. These results indicate that our detection scheme is able to reliably detect additive noise in images previously JPEG-compressed using a camera's default settings.

Additionally, we evaluated our additive noise detection technique's ability to operate on images previously JPEG-compressed at different quality factors. To do this, we JPEG-compressed each of the 244 images in the Uncompressed Colour Image Database at the quality factors $Q = 90$, 70, 50, and 30 [21]. As before, we created a set of altered images by adding unit variance Gaussian noise to each image, then saved each image as a bitmap. We then tested each image for the presence of additive noise with our proposed forensic technique using a variety of detection thresholds. We conducted this experiment using the same experimental parameters as our previous test. For each threshold, the probabilities of detection and false alarm were calculated then used to construct the series of ROC curves displayed in Fig. 15. Results comparable to our previous experiment were achieved for images previously compressed using quality factors of 50 or greater. For these quality factors, a $P_{\rm d}$ of 99% was acheived at a $P_{\rm fa}$ of 3.7% or less. At lower quality factors, however, noise detection appears to become more difficult.

VI. Conclusion

In this paper, we proposed a set of digital image forensic techniques capable of detecting global and local contrast enhancement, identifying the use of histogram equalization, and detection of the global addition of noise to a previously JPEG-compressed image. In each of these techniques, detection depends upon the presence or absence of an intrinsic fingerprint introduced into an image's histogram by a pixel value mapping.

We developed a model of an unaltered image's pixel value histogram and provided justification for this model. We defined the intrinsic fingerprint which a mapping leaves in the histogram of of an image's pixel values or other discrete valued data. By observing that the intrinsic fingerprints of contrast enhancement operations add energy to the high frequency components of an image's pixel value histogram, we developed a global contrast enhancement detection technique. We extended this technique into a method for detecting locally applied contrast enhancement and demonstrated its usefulness for detecting cut and paste type forgeries. Characteristic features of histogram equalization's intrinsic fingerprint were identified and used to propose a scheme for identifying the use of this operation. Additionally, we proposed a technique which detects the global addition of noise to a previously JPEG-compressed image by searching for the intrinsic fingerprint of a specific pixel value mapping applied to the image in question.

Through detailed simulations, we tested the effectiveness of each of the proposed forensic techniques. Our simulation results show that aside from exceptional cases, each of the proposed techniques achieved a $P_{\rm d}$ of 99% with a $P_{\rm fa}$ of 7% or less. These

results indicate that all of the proposed forensic techniques are very useful tools for identifying image manipulations and forgeries.

REFERENCES

[1] M. Chen, J. Fridrich, M. Goljan, and J. Lukáš, "Determining image origin and integrity using sensor noise," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 1, pp. 74–90, Mar. 2008.

[2] T.-T. Ng, S.-F. Chang, J. Hsu, L. Xie, and M. P. Tsui, "Physics-motivated features for distinguishing photographic images and computer graphics," in *Proc. ACM Multimdedia*, Singapore, 2005, pp. 239–248.

[3] 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, Sep. 2007.

[4] M. K. Johnson and H. Farid, "Exposing digital forgeries by detecting inconsistencies in lighting," in *Proc. ACM Multimedia and Security Workshop*, New York, NY, 2005, pp. 1–10.

[5] M. K. Johnson and H. Farid, "Exposing digital forgeries through chromatic aberration," in *Proc. ACM Multimedia and Security Workshop*, Geneva, Switzerland, 2006, pp. 48–55.

[6] A. C. Popescu and H. Farid, "Exposing digital forgeries in color filter array interpolated images," *IEEE Trans. Signal Process.*, vol. 53, no. 10, pp. 3948–3959, Oct. 2005.

[7] T.-T. Ng, S.-F. Chang, and Q. Sun, "Blind detection of photomontage using higher order statistics," in *Proc. IEEE Int. Symp. Circuits Systems*, Vancouver, BC, Canada, May 2004, vol. 5, pp. V-688–V-691.

[8] S. Bayram, I. Avcibas, B. Sankur, and N. Memon, "Image manipulation detection," *J. Electron. Imag.*, vol. 15, no. 4, p. 041102, 2006.

[9] I. Avcibas, S. Bayram, N. Memon, M. Ramkumar, and B. Sankur, "A classifier design for detecting image manipulations," in *Proc. ICIP*, Oct. 2004, vol. 4, pp. 2645–2648.

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

[11] J. Lukáš, J. Fridrich, and M. Goljan, "Detecting digital image forgeries using sensor pattern noise," in *Proc. SPIE, Electronic Imaging, Security, Steganography, Watermarking of Multimedia Contents*, San Jose, CA, Feb. 2006, vol. 6072, pp. 362–372.

[12] A. Swaminathan, M. Wu, and K. J. R. Liu, "Digital image forensics via intrinsic fingerprints," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 1, pp. 101–117, Mar. 2008.

[13] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Trans. Signal Process.*, vol. 53, pp. 758–767, Feb. 2005.

[14] A. C. Popescu and H. Farid, "Statistical tools for digital forensics," in *Proc. 6th Int. Workshop on Information Hiding*, Toronto, Canada, 2004, pp. 128–147.

[15] T. Pevný and J. Fridrich, "Detection of double-compression in JPEG images for applications in steganography," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 2, pp. 247–258, Jun. 2008.

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

[17] H. Farid, "Blind inverse gamma correction," *IEEE Trans. Image Process.*, vol. 10, pp. 1428–1433, Oct. 2001.

[18] J. Fridrich, D. Soukal, and J. Lukáš, "Detection of copy-move forgery in digital images," in *Proc. Digital Forensic Research Workshop*, Cleveland, OH, 2003.

[19] G. E. Healey and R. Kondepudy, "Radiometric CCD camera calibration and noise estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 3, pp. 267–276, Mar. 1994.

[20] M. Stamm and K. J. R. Liu, "Blind forensics of contrast enhancement in digital images," in *Proc. ICIP*, San Diego, CA, Oct. 2008, pp. 3112–3115.

[21] G. Schaefer and M. Stich, "UCID: An uncompressed color image database," in *Proc. SPIE: Storage and Retrieval Methods and Applications for Multimedia*, 2003, vol. 5307, pp. 472–480.

[22] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Boston, MA: Addison-Wesley, 2001.

[23] M. C. Stamm and K. J. R. Liu, "Forensic detection of image tampering using intrinsic statistical fingerprints in histograms," in *Proc. APSIPA Annual Summit and Conf.*, Sapporo, Japan, Oct. 2009.

Matthew C. Stamm (S'08) received the B.S. degree in electrical engineering from the University of Maryland, College Park, in 2004. He is currently working toward the Ph.D. degree at the Department of Electrical and Computer Engineering, University of Maryland, College Park.

From 2004 to 2006, he was a radar systems engineer at the Johns Hopkins University Applied Physics Laboratory. His current research interests include digital multimedia forensics and anti-forensics as well as music information retrieval.

Mr. Stamm received a Distinguished Teaching Assistant Award in 2006 as well as a Future Faculty Fellowship in 2010 from the University of Maryland.

K. J. Ray Liu (S'87–M'90–SM'93–F'03) is a Distinguished Scholar-Teacher of University of Maryland, College Park. He is Associate Chair, Graduate Studies and Research, Electrical and Computer Engineering Department, and leads the Maryland Signals and Information Group conducting research encompassing broad aspects of information science and technology including communications and networking, information forensics and security, multimedia signal processing, and biomedical engineering.

Dr. Liu is the recipient of numerous honors and awards, including best paper awards from the IEEE Signal Processing Society, the IEEE Vehicular Technology Society, and EURASIP; an IEEE Signal Processing Society Distinguished Lecturer, the EURASIP Meritorious Service Award, and the National Science Foundation Young Investigator Award. He also received various teaching and research recognitions from University of Maryland, including the 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. He is a Fellow of AAAS. He is President-Elect and was Vice President–Publications of the IEEE Signal Processing Society. He was the Editor-in-Chief of the *IEEE Signal Processing Magazine* and the founding Editor-in-Chief of the *EURASIP Journal on Applied Signal Processing*. His recent books include *Behavior Dynamics in Media-Sharing Social Networks* (Cambridge Univ. Press, to be publihsed); *Cognitive Radio Networking and Security: A Game Theoretical View* (Cambridge Univ. Press, 2010; *Handbook on Array Processing and Sensor Networks* (IEEE-Wiley, 2009); *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).