Statistical Tools for Digital Forensics

Alin C. Popescu and Hany Farid *

Department of Computer Science at Dartmouth College

Abstract. A digitally altered photograph, often leaving no visual clues of having been tampered with, can be indistinguishable from an authentic photograph. As a result, photographs no longer hold the unique stature as a definitive recording of events. We describe several statistical techniques for detecting traces of digital tampering in the absence of any digital watermark or signature. In particular, we quantify statistical correlations that result from specific forms of digital tampering, and devise detection schemes to reveal these correlations.

1 Introduction

The advent of low-cost and high-resolution digital cameras, and sophisticated photo-editing software, has made it remarkably easy to manipulate and alter digital images. In addition, digital forgeries, often leaving no visual clues of having been tampered with, can be indistinguishable from authentic photographs. And while the technology to manipulate digital media is developing at break-neck speeds, the technology to contend with its ramifications is lagging behind.

Digital watermarking has been proposed as a means by which an image can be authenticated (see, for example, [12,3] for general surveys). Within this broad area, several authentication schemes have been proposed: embedded signatures [10,24,25,18,2], erasable fragile watermarks [11,9], semi-fragile watermarks [16,23,28,15], robust tell-tale watermarks [27,14,28], and self-embedding watermarks [8]. All of these approaches work by either inserting at the time of recording an imperceptible digital code (a watermark) into the image, or extracting at the time of recording a digital code (a signature) from the image and re-inserting it into the image. With the assumption that tampering will alter a watermark, an image can be authenticated by verifying that the extracted watermark is the same as that which was inserted. The major drawback of this approach is that a watermark must be inserted at precisely the time of recording, which would limit this approach to specially equipped digital cameras. This method also relies on the assumption that the watermark cannot be easily removed and reinserted — it is not yet clear whether this is a reasonable assumption (e.g., [4]).

In contrast to these approaches, we describe a class of statistical techniques for detecting traces of digital tampering in the absence of any watermark or signature. These approaches work on the assumption that although digital forgeries may leave no visual clues of having been tampered with, they may, nevertheless, alter the underlying statistics of an image. Consider, for example, the creation of a digital forgery that shows a pair of famous movie stars, rumored to have a romantic relationship, walking hand-in-hand. Such a photograph could be created by splicing together individual images of each movie star and overlaying the digitally created composite onto a sunset beach. In order to create a convincing match, it is often necessary to (1) re-size, rotate, or stretch portions of the images; (2) apply luminance non-linearities (e.g., gamma correction) to portions of the image in order to adjust for brightness differences; (3) add small amounts of noise to conceal evidence of tampering; and (4) re-save the final image (typically with lossy compression such as JPEG). Although these manipulations are often imperceptible to the human eye, they may introduce specific correlations into the image, which when detected can be used as evidence of digital tampering. In this paper, we quantify statistical correlations that result from each of these specific forms of digital tampering, and devise detection schemes to reveal the correlations. The effectiveness of these techniques is shown on a number of simple synthetic examples and on perceptually credible forgeries.

* 6211 Sudikoff Lab, Dartmouth College, Hanover NH 03755 USA (email: farid@cs.dartmouth.edu; tel/fax: 603.646.2761/603.646.1672). This work was supported by an Alfred P. Sloan Fellowship, an NSF CAREER Award (IIS-99-83806), an NSF Infrastructure Grant (EIA-98-02068), and under Award No. 2000-DT-CX-K001 from the Office for Domestic Preparedness, U.S. Department of Homeland Security (points of view in this document are those of the author(s) and do not necessarily represent the official position of the U.S. Department of Homeland Security).

2 Re-sampling

Consider the scenario in which a digital forgery is created by splicing together two, or more, individual images. In order to create a convincing match, it is often necessary to re-size, rotate, or stretch the images, or portions of them. These manipulations require re-sampling an image onto a new sampling lattice using some form of interpolation. Although, the re-sampling of an image is often imperceptible, specific correlations are introduced in the re-sampled image. When detected, these correlations represent evidence of tampering. We describe the form of these correlations, and propose an algorithm for detecting them in any portion of an image.

For purposes of exposition we will first describe how and where re-sampling introduces correlations in 1-D signals, and how to detect these correlations. The relatively straight-forward generalization to 2-D images is then presented.

2.1 Re-sampling Signals

Consider a 1-D discretely-sampled signal $x[t]$ with m samples. The number of samples in this signal can be increased or decreased by a factor p/q to n samples in three steps [21]:

1. up-sample: create a new signal $x_u[t]$ with pm samples, where $x_u[pt] = x[t]$, $t = 1, 2, ..., m$, and $x_u[t] = 0$ otherwise.
2. interpolate: convolve $x_u[t]$ with a low-pass filter: $x_i[t] = x_u[t] \star h[t]$.
3. down-sample: create a new signal $x_d[t]$ with n samples, where $x_d[t] = x_i[qt]$, $t = 1, 2, ..., n$. Denote the re-sampled signal as $y[t] \equiv x_d[t]$.

Different types of re-sampling algorithms (e.g., linear, cubic) differ in the form of the interpolation filter $h[t]$ in step 2. Since all three steps in the re-sampling of a signal are linear, this process can be described with a single linear equation. Denoting the original and re-sampled signals in vector form, $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively, re-sampling takes the form: $\boldsymbol{y} = A_{p/q}\boldsymbol{x}$, where the $n \times m$ matrix $A_{p/q}$ embodies the entire re-sampling process. For example, the matrices for up-sampling by a factor of $4/3$ and $2/1$ using linear interpolation have the form:

$$A_{4/3} = \begin{bmatrix} 1 & 0 & 0 & 0 & \\ 0.25 & 0.75 & 0 & 0 & \\ 0 & 0.50 & 0.50 & 0 & \\ 0 & 0 & 0.75 & 0.25 & \\ 0 & 0 & 0 & 1 & \\ & & & & \ddots \end{bmatrix}, \qquad A_{2/1} = \begin{bmatrix} 1 & 0 & 0 & \\ 0.5 & 0.5 & 0 & \\ 0 & 1 & 0 & \\ 0 & 0.5 & 0.5 & \\ 0 & 0 & 1 & \\ & & & \ddots \end{bmatrix}. \tag{1}$$

Depending on the re-sampling rate, the re-sampling process will introduce correlations of varying degrees between neighboring samples. For example, consider the up-sampling of a signal by a factor of two using linear interpolation. Here, the odd samples of the re-sampled signal $\boldsymbol{y}$ take on the values of the original signal $\boldsymbol{x}$, i.e., $y_{2i-1} = x_i$, $i = 1, \ldots, m$. The even samples, on the other hand, are the average of adjacent neighbors of the original signal: $y_{2i} = 0.5x_i + 0.5x_{i+1}$, where $i = 1, \ldots, m-1$. Note that since each sample of the original signal can be found in the re-sampled signal, i.e., $x_i = y_{2i-1}$ and $x_{i+1} = y_{2i+1}$, the above relationship can be expressed in terms of the re-sampled samples only: $y_{2i} = 0.5y_{2i-1} + 0.5y_{2i+1}$. That is, across the entire re-sampled signal, each even sample is precisely the same linear combination of its adjacent two neighbors. In this simple case, at least, a re-sampled signal could be detected (in the absence of noise) by noticing that every other sample is perfectly correlated to its neighbors. To be useful in a general forensic setting we need, at a minimum, for these types of correlations to be present regardless of the re-sampling rate.

Consider now re-sampling a signal by an arbitrary amount p/q. In this case we first ask, when is the i^{th} sample of a re-sampled signal equal to a linear combination of its $2N$ neighbors, that is:

$$y_i \stackrel{?}{=} \sum_{k=-N}^{N} \alpha_k y_{i+k}, \tag{2}$$

where α_k are scalar weights (and $\alpha_0 = 0$). Re-ordering terms, and re-writing the above constraint in terms of the re-sampling matrix yields:

$$y_i - \sum_{k=-N}^{N} \alpha_k y_{i+k} = 0 \quad \Rightarrow \quad (\boldsymbol{a}_i \cdot \boldsymbol{x}) - \sum_{k=-N}^{N} \alpha_k (\boldsymbol{a}_{i+k} \cdot \boldsymbol{x}) = 0 \quad \Rightarrow \quad \left(\boldsymbol{a}_i - \sum_{k=-N}^{N} \alpha_k \boldsymbol{a}_{i+k} \right) \cdot \boldsymbol{x} = 0, \tag{3}$$

where $\boldsymbol{a}_i$ is the i^{th} row of the re-sampling matrix $A_{p/q}$, and $\boldsymbol{x}$ is the original signal. We see now that the i^{th} sample of a re-sampled signal is equal to a linear combination of its neighbors when the i^{th} row of the re-sampling matrix, $\boldsymbol{a}_i$, is equal to a linear combination of its neighboring rows, $\sum_{k=-N}^{N} \alpha_k \boldsymbol{a}_{i+k}$. For example, in the case of up-sampling by a factor of two ($A_{2/1}$ in Equation (1)), the even rows are a linear combination of the two adjacent odd rows. Note also that if the i^{th} sample is a linear combination of its neighbors then the $(i - kp)^{th}$ sample (k an integer) will be the same combination of its neighbors, that is, the correlations are periodic. It is, of course, possible for the constraint of Equation (3) to be satisfied when the difference on the left-hand side of the equation is orthogonal to the original signal $\boldsymbol{x}$. While this may occur on occasion, these correlations are unlikely to be periodic.

2.2 Detecting Re-sampling

Given a signal that has been re-sampled by a known amount and interpolation method, it is possible to find a set of periodic samples that are correlated in the same way to their neighbors. For example, consider the re-sampling matrix, $A_{4/3}$, of Equation (1). Here, based on the periodicity of the re-sampling matrix, we see that, for example, the 3^{rd}, 7^{th}, 11^{th}, etc. samples of the re-sampled signal will have the same correlations to their neighbors. The specific form of the correlations can be determined by finding the neighborhood size, N, and the set of weights, $\boldsymbol{\alpha}$, that satisfy: $\boldsymbol{a}_i = \sum_{k=-N}^{N} \alpha_k \boldsymbol{a}_{i+k}$, Equation (3), where $\boldsymbol{a}_i$ is the i^{th} row of the re-sampling matrix and $i = 3, 7, 11$, etc. If, on the other-hand, we know the specific form of the correlations, $\boldsymbol{\alpha}$, then it is straight-forward to determine which samples satisfy $y_i = \sum_{k=-N}^{N} \alpha_k y_{i+k}$, Equation (3).

In practice, of course, neither the re-sampling amount nor the specific form of the correlations are typically known. In order to determine if a signal has been re-sampled, we employ the expectation/maximization algorithm (EM) [5] to simultaneously estimate a set of periodic samples that are correlated to their neighbors, and the specific form of these correlations. We begin by assuming that each sample belongs to one of two models. The first model, M_1, corresponds to those samples that are correlated to their neighbors, and the second model, M_2, corresponds to those samples that are not (i.e., an outlier model). The EM algorithm is a two-step iterative algorithm: (1) in the E-step the probability that each sample belongs to each model is estimated; and (2) in the M-step the specific form of the correlations between samples is estimated. More specifically, in the E-step, the probability of each sample, y_i, belonging to model M_1 is given by Bayes' rule:

$$\Pr\{y_i \in M_1 \mid y_i\} = \frac{\Pr\{y_i \mid y_i \in M_1\} \Pr\{y_i \in M_1\}}{\Pr\{y_i \mid y_i \in M_1\} \Pr\{y_i \in M_1\} + \Pr\{y_i \mid y_i \in M_2\} \Pr\{y_i \in M_2\}}, \tag{4}$$

where equal priors are assumed, i.e., $\Pr\{y_i \in M_1\} = \Pr\{y_i \in M_1\} = 1/2$. We also assume that:

$$\Pr\{y_i \mid y_i \in M_1\} = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-\left(y_i - \sum_{k=-N}^{N} \alpha_k y_{i+k} \right)^2}{2\sigma^2} \right], \tag{5}$$

and that $\Pr\{y_i \mid y_i \in M_2\}$ is uniformly distributed over the range of possible values of the signal $\boldsymbol{y}$. The variance, σ, of the above Gaussian distribution is estimated in the M-step. Note that the E-step requires an estimate of $\boldsymbol{\alpha}$, which on the first iteration is chosen randomly. In the M-step, a new estimate of $\boldsymbol{\alpha}$ is computed using weighted least-squares, that is, minimizing the following quadratic error function:

$$E(\boldsymbol{\alpha}) = \sum_i w(i) \left(y_i - \sum_{k=-N}^{N} \alpha_k y_{i+k} \right)^2, \tag{6}$$

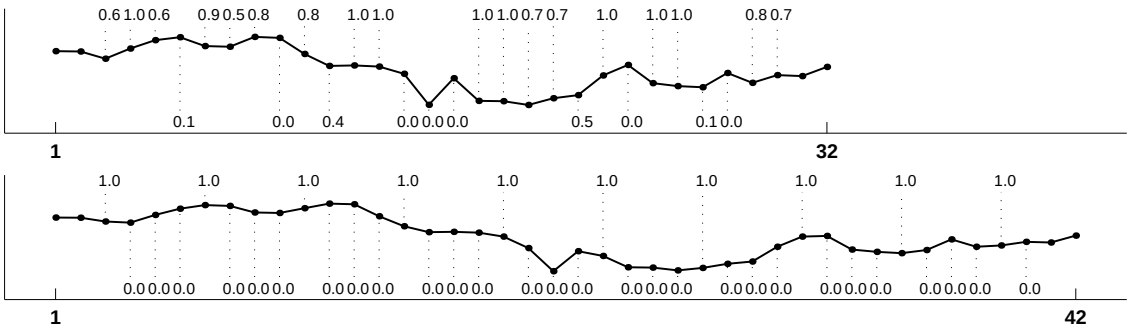

Fig. 1: A signal with 32 samples (top) and this signal re-sampled by a factor of $4/3$ (bottom). Each sample is annotated with its probability of being correlated to its neighbors. Note that for the up-sampled signal these probabilities are periodic, while for the original signal they are not.

where the weights $w(i) \equiv \Pr\{y_i \in M_1 \mid y_i\}$, Equation (4), and $\alpha_0 = 0$. This error function is minimized by computing the gradient with respect to $\boldsymbol{\alpha}$, setting the result equal to zero, and solving for $\boldsymbol{\alpha}$, yielding:

$$\boldsymbol{\alpha} = (Y^T W Y)^{-1} Y^T W \boldsymbol{y}, \tag{7}$$

where the i^{th} row of the matrix Y is given by: $\begin{bmatrix} y_i & \dots & y_{N+i-1} & y_{N+i+1} & \dots & y_{2N+i} \end{bmatrix}$, and W is a diagonal weighting matrix with $w(i)$ along the diagonal. The E-step and M-step are iteratively executed until a stable estimate of $\boldsymbol{\alpha}$ is achieved.

Shown in Fig. 1 are the results of running EM on an original and re-sampled (by a factor of $4/3$) signal. Shown on the top is the original signal where each sample is annotated with its probability of being correlated to its neighbors (the first and last two samples are not annotated due to border effects — a neighborhood size of five ($N = 2$) was used in this example). Similarly, shown on the bottom is the re-sampled signal and the corresponding probabilities. In the latter case, the periodic pattern is obvious, where only every 4^{th} sample has probability 1, as would be expected by an up-sampling by a factor of $4/3$, Equation (1). As expected, no periodic pattern is present in the original signal.

The periodic pattern introduced by re-sampling depends, of course, on the re-sampling rate. As a result, it is possible to not only uncover traces of re-sampling, but to also estimate the amount of re-sampling. It is not possible, however, to uniquely determine the specific amount of re-sampling as there are re-sampling parameters that yield similar periodic patterns. [1] There is also a range of re-sampling rates that will not introduce periodic correlations. For example, consider down-sampling by a factor of two (for simplicity, consider the case where there is no interpolation, i.e., $y_i = x_{2i}$). In this case, the rows of the re-sampling matrix are orthogonal to one another, and as a result no row can be written as a linear combination of its neighboring rows. In general, the detectability of any re-sampling can be estimated by generating the re-sampling matrix and determining if neighboring rows are linearly dependent.

2.3 Re-sampling Images

In the previous sections we showed that for 1-D signals re-sampling introduces periodic correlations and that these correlations can be detected using the EM algorithm. The extension to 2-D images is relatively straightforward. As with 1-D signals, the up-sampling or down-sampling of an image is still linear and involves the same three steps: up-sampling, interpolation, and down-sampling — these steps are simply carried out on a 2-D lattice. Again, as with 1-D signals, the re-sampling of an image introduces periodic correlations. Consider, for example, the simple case of up-sampling by a factor of two using linear interpolation. In the re-sampled image, the pixels in odd rows and even columns will be the average of their two closest horizontal neighbors, while the pixels in even rows and odd columns will be the average of their two closest vertical neighbors. That is, the correlations are, as with the 1-D signals, periodic. And in the same way that EM was used to uncover periodic correlations in 1-D signals, the same approach can be used with 2-D images.

[1] In general, two re-sampling rates p_1/q_1 and p_2/q_2 will generate similar periodic patterns if either $\{p_1/q_1\} = \{p_2/q_2\}$, or $\{p_1/q_1\} = 1 - \{p_2/q_2\}$, where $\{\cdot\}$ denotes the fractional part of a number.

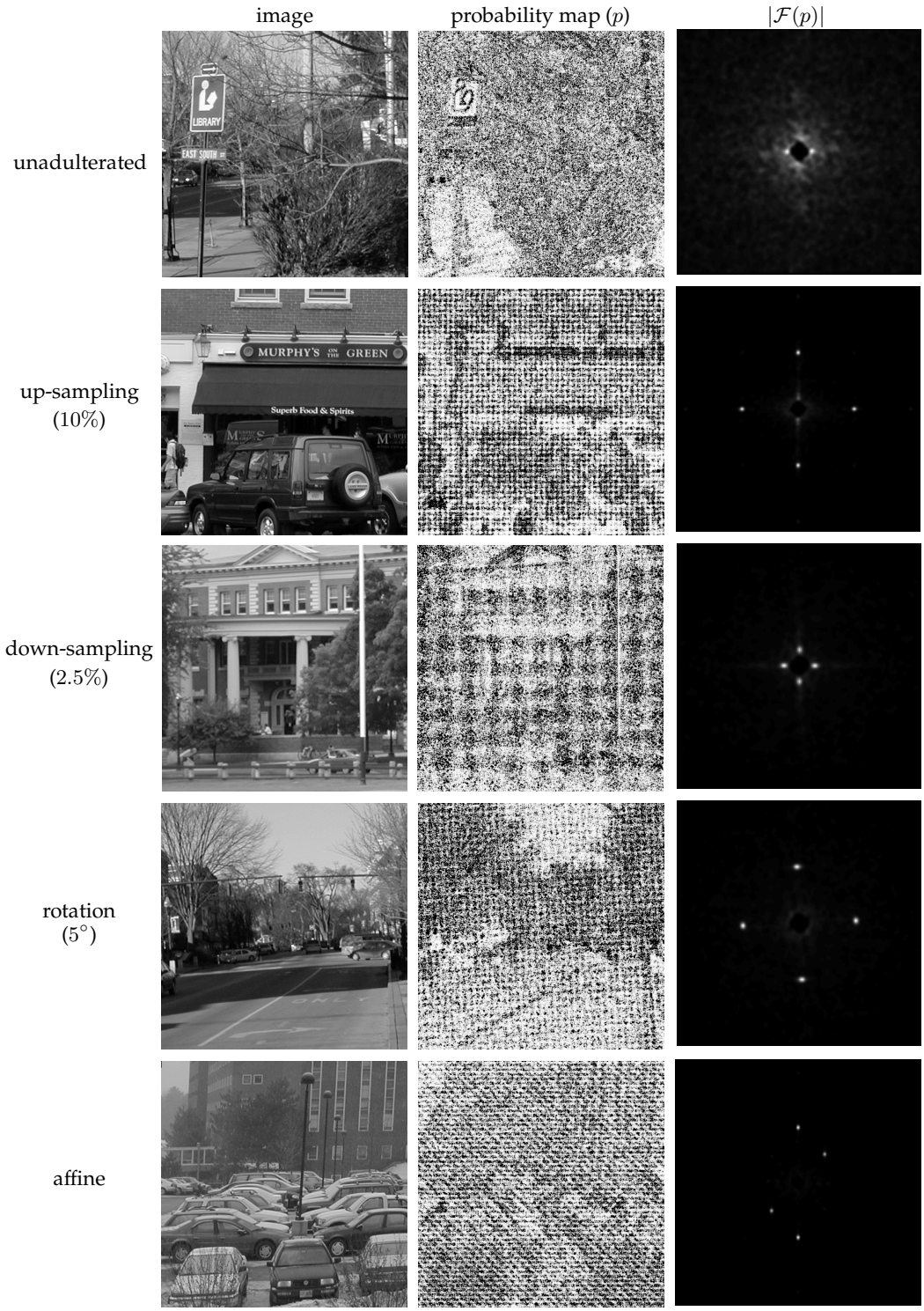

Fig. 2: Shown in the top row is an unadulterated image, and shown below are images re-sampled with different parameters. Shown in the middle column are the estimated probability maps that embody the spatial correlations in the image. The magnitude of the Fourier transforms of these maps are shown in the right-most column. Note that only the re-sampled images yield periodic maps.

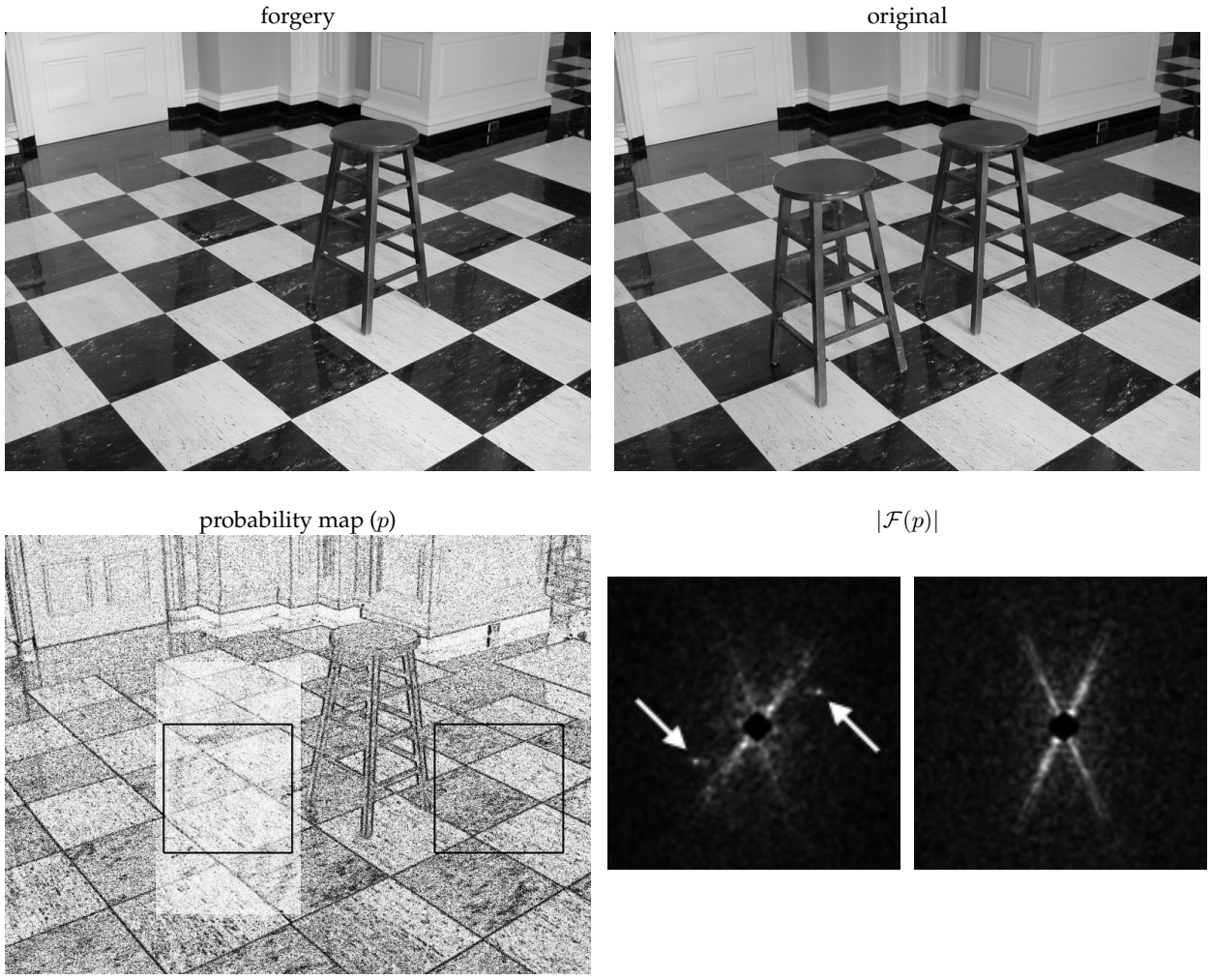

Fig. 3: Shown along the top row is a forgery and the original image. The forgery consists of removing a stool and splicing in a new floor taken from another image (not shown here) of the same room. Shown below is the estimated probability map (p) of the forgery, and the magnitude of the Fourier transform of a region in the new floor (left) and on the original floor (right). The periodic pattern (spikes in $|\mathcal{F}(p)|$) in the new floor suggest that this region was re-sampled.

2.4 Results

For the results presented here, we built a database of 200 grayscale images in TIFF format. These images were 512×512 pixels in size. Each of these images were cropped from a smaller set of twenty-five 1200×1600 images taken with a Nikon Coolpix 950 camera (the camera was set to capture and store in uncompressed TIFF format). Using bi-cubic interpolation these images were up-sampled, down-sampled, rotated, or affine transformed by varying amounts. Although we will present results for grayscale images, the generalization to color images is straight-forward — each color channel would be independently subjected to the same analysis as that described below.

For the original and re-sampled images, the EM algorithm described in Section 2.2 was used to estimate probability maps that embody the correlation between each pixel and its neighbors. The neighborhood size was fixed throughout to be 5×5. Shown in Fig. 2 are several examples of the periodic patterns that emerged due to re-sampling. In the top row of the figure are (from left to right) the original unadulterated image, the estimated probability map and the magnitude of the central portion of the Fourier transform of this map (for display purposes, each Fourier transform was independently auto-scaled to fill the full intensity range and

high-pass filtered to remove the lowest frequencies). Shown below this row are images uniformly re-sampled (using bi-cubic interpolation) with different parameters. For the re-sampled images, note the periodic nature of their probability maps and the corresponding localized peaks in their Fourier transforms.

Shown in Fig. 3 is an example of our detection algorithm applied to an image where only a portion of the image was re-sampled. That is, the forged image contains a region that was re-sampled (up-sampled, rotated, and non-linearly distorted). Shown are the original photograph, the forgery, and the estimated probability map. Note that the re-sampled region is clearly detected - while the periodic pattern is not particularly visible in the spatial domain at the reduced scale, the well localized peaks in the Fourier domain clearly reveal its presence (for display purposes, the Fourier transform was auto-scaled to fill the full intensity range and high-pass filtered to remove the lowest frequencies).

It may seem, at first glance, that the detection of re-sampling correlations will be sensitive to simple counter-attacks — for example, small amounts additive noise. We have found, however, that due to the global nature of the EM estimation, the correlations can be detected even in the presence of additive noise and luminance non-linearities (e.g., gamma correction). A full exploration of the robustness is beyond the scope of this paper.

3 Double JPEG Compression

Tampering with a digital image requires the use of a photo-editing software such as Adobe PhotoShop. In the making of digital forgeries an image is loaded into the editing software, some manipulations are performed, and the image is re-saved. Since most images are stored in JPEG format (e.g., a majority of digital cameras store images directly in JPEG format), it is likely that both the original and forged images are stored in this format. Notice that in this scenario the forged image is double JPEG compressed. Double JPEG compression introduces specific artifacts not present in singly compressed images (this observation has also been noted in [17]). Note that evidence of double JPEG compression, however, does not necessarily prove malicious tampering. For example, it is possible for a user to simply re-save a high quality JPEG image with a lower quality. The authenticity of a double JPEG compressed image should, however, be called into question. We start by giving a short description of the JPEG compression algorithm and then quantify the artifacts introduced by double compression.

3.1 JPEG Compression

JPEG is a standardized image compression procedure proposed by a committee with the same name JPEG (Joint Photographic Experts Committee). To be generally applicable, the JPEG standard [1] specified two compression schemes: a lossless predictive scheme and a lossy scheme based on the Discrete Cosine Transform (DCT). The most popular lossy compression technique is known as the baseline method and encompasses a subset of the DCT-based modes of operation. The encoding of an image involves three basic steps [26]:

1. Discrete Cosine Transform (DCT): An image is divided into 8×8 blocks in raster scan order (left to right, top to bottom), shifted from unsigned to signed integers (e.g., from $[0, 255]$ to $[-128, 127]$), and each block's DCT computed.
2. Quantization: The DCT coefficients obtained in the previous step are uniformly quantized, i.e., divided by a quantization step and rounded off to the nearest integer. Since quantization is a non-invertible operation this step represents the main source of information loss.
3. Entropy Encoding: This step involves lossless entropy compression that transforms the quantized DCT coefficients into a stream of compressed data. The most frequently used procedure is Huffman coding, although arithmetic coding is also supported.

The decoding of a compressed data stream involves the inverse of the previous three steps, taken in reverse order: entropy decoding, de-quantization, and inverse DCT.

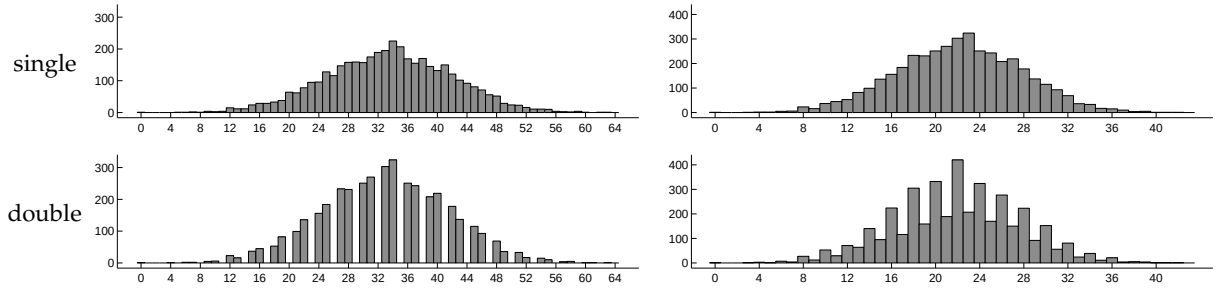

Fig. 4: Shown along the top row are histograms of single quantized signals with steps 2 (left) and 3 (right). Shown in the bottom row are histograms of double quantized signals with steps 3 followed by 2 (left), and 2 followed by 3 (right). Note the periodic artifacts in the histograms of double quantized signals.

3.2 Double Quantization

Consider the example of a generic discrete 1-D signal $x[t]$. Quantization is a point-wise operation that is described by a one-parameter family of functions: [2]

$$q_a(u) = \left\lfloor \frac{u}{a} \right\rfloor, \tag{8}$$

where a is the quantization step (a strictly positive integer), and u denotes a value in the range of $x[t]$. De-quantization brings the quantized values back to their original range: $q_a^{-1}(u) = au$. Note that the function $q_a(u)$ is not invertible, and that de-quantization is not the inverse function of quantization. Double quantization is a point-wise operation described by a two-parameter family of functions:

$$q_{ab}(u) = \left\lfloor \left\lfloor \frac{u}{b} \right\rfloor \frac{b}{a} \right\rfloor, \tag{9}$$

where a and b are the quantization steps (strictly positive integers). Notice that double quantization can be represented as a sequence of three steps: quantization with step b, followed by de-quantization with step b, followed by quantization with step a.

Consider an example where the samples of $x[t]$ are normally distributed in the range $[0, 127]$. To illustrate the nature of the double quantization artifacts, we quantize the signal $x[t]$ in four different ways, and show the resulting histograms, Fig. 4. Shown along the top row of this figure are the histograms of the same signal quantized with steps 2 and 3. Shown in the bottom row are the histograms of the same signal double quantized with steps 3 followed by 2, and 2 followed by 3. When the step size decreases (bottom left) some bins in the histogram are empty. This is not surprising since the first quantization places the samples of the original signal into 42 bins, while the second quantization re-distributes them into 64 bins. When the step size increases (bottom right) some bins contain more samples than their neighboring bins. This also is to be expected since the even bins receive samples from four original histogram bins, while the odd bins receive samples from only two. In both cases of double quantization, note the periodicity of the artifacts introduced into the histograms.

To better understand why the double quantization of a signal introduces periodic artifacts, we will analyze the dependence between the histograms of single and double quantized signals. Consider first the case of a single quantized signal denoted by $x_a[t] = q_a(x[t])$, and denote the histograms of the original and quantized signals by $H(u)$ and $H_a(v)$. Since $q_a(\cdot)$ is a many-to-one function, several values from the range of $x[t]$ will map onto the same value in the range of $x_a[t]$, i.e., several bins from H contribute to a bin in H_a. For example, let v denote a value in the range of $x_a[t]$, then the values in the range of $x[t]$ that map to it are in the range $[av, av + (a - 1)]$. Therefore, the relationship between $H(u)$ and $H_a(v)$ is given by: $H_a(v) = \sum_{k=0}^{a-1} H(av + k)$. Note that there are exactly a bins in the original histogram that contribute to each bin in the histogram of the quantized signal. Consider next the case of a double quantized signal denoted by $x_{ab}[t] = q_{ab}(x[t])$, and let its histogram be denoted by $H_{ab}(v)$. In contrast to the single quantization case, the number of bins of H that

[2] For the purpose of illustration and in order to make the analysis easier we will use the floor function in the quantization function. Similar results can be shown if integer rounding is used instead.

contribute to a bin of H_{ab} will depend on the double quantized bin value. Let v be a value in the range of $x_{ab}[t]$. Denote u_{min} and u_{max} as the smallest and largest values of u in the range of $x[t]$ that map to v, that is, they satisfy the following:

$$\left\lfloor \left\lfloor \frac{u}{b} \right\rfloor \frac{b}{a} \right\rfloor = v. \tag{10}$$

Using the following property of the floor function:

$$\lfloor z \rfloor = m \quad \Rightarrow \quad m \leq z < m+1, \tag{11}$$

where z is an arbitrary real number and m an integer, Equation (10) implies:

$$v \leq \left\lfloor \frac{u}{b} \right\rfloor \frac{b}{a} < v+1 \quad \Leftrightarrow \quad \frac{a}{b}v \leq \left\lfloor \frac{u}{b} \right\rfloor < \frac{a}{b}(v+1). \tag{12}$$

Since $\lfloor u/b \rfloor$ is an integer , Equation (12) can be rewritten using the ceiling function to include only integers:

$$\left\lceil \frac{a}{b}v \right\rceil \leq \left\lfloor \frac{u}{b} \right\rfloor \leq \left\lceil \frac{a}{b}(v+1) \right\rceil - 1. \tag{13}$$

From Equation (13) it can be seen that u_{min} must satisfy:

$$\left\lfloor \frac{u_{min}}{b} \right\rfloor = \left\lceil \frac{a}{b}v \right\rceil \quad \Rightarrow \quad u_{min} = \left\lceil \frac{a}{b}v \right\rceil b, \tag{14}$$

while u_{max} must satisfy:

$$\left\lfloor \frac{u_{max}}{b} \right\rfloor = \left\lceil \frac{a}{b}(v+1) \right\rceil - 1 \quad \Rightarrow \quad u_{max} = \left(\left\lceil \frac{a}{b}(v+1) \right\rceil - 1 \right) b + (b-1) = \left\lceil \frac{a}{b}(v+1) \right\rceil b - 1. \tag{15}$$

Since double quantization is a monotonically increasing function, it follows that all the values between u_{min} and u_{max} will map to v through double quantization. The relationship between the original and double quantized histogram then takes the form:

$$H_{ab}(v) = \sum_{u=u_{min}}^{u_{max}} H(u). \tag{16}$$

Note that the number of original histogram bins, $n(v)$, contributing to bin v in the double quantized histogram depends on v, and from Equations (14) and (15), can be expressed as:

$$n(v) = u_{max} - u_{min} + 1 \quad = \quad b \left(\left\lceil \frac{a}{b}(v+1) \right\rceil - \left\lceil \frac{a}{b}v \right\rceil \right). \tag{17}$$

Note that $n(v)$ is a periodic function with period b, i.e., $n(v) = n(v+b)$. This periodicity is the reason periodic artifacts appear in histograms of double quantized signals.

From Equation (17), the double quantization artifacts shown in Fig. 4 can now be explained. Consider first the case of double quantization using steps $b = 3$ followed by $a = 2$, (bottom-left panel in Fig. 4). The number of original histogram bins contributing to double quantized histogram bins of the form $(3k+2)$ (k integer) is given by:

$$n(3k+2) \quad = \quad 3 \left(\left\lceil \frac{2}{3}(3k+3) \right\rceil - \left\lceil \frac{2}{3}(3k+2) \right\rceil \right) \quad = \quad 3 \left(2k+2-2k - \left\lceil \frac{4}{3} \right\rceil \right) \quad = \quad 0. \tag{18}$$

This is consistent with the observation that every $(3k+2)^{nd}$ (k integer) bin of the double quantized histogram is empty. In the second example of double quantization in Fig. 4, $b = 2$ and $a = 3$, it can be shown that $n(2k) = 4$ and $n(2k+1) = 2$ (k integer). Again, this is consistent with the periodic artifacts shown in the bottom-right panel of Fig. 4.

There are cases when the histogram of a double quantized signal does not contain periodic artifacts. For example, if in Equation (17) a/b is an integer then $n(v) = a$. Note that the same result is obtained if the signal were single quantized with step a. In this case, single and double quantization of a signal yields the same histogram, therefore it is impossible to distinguish between the two. Notice also in Equation (16) that the histogram of the double quantized signal, H_{ab}, depends on the values of the histogram of the original signal H. It is conceivable that histograms of original signals may contain naturally occuring artifacts that could mask those introduced by double quantization. While this may happen on occasion, such artifacts do not occur often.

3.3 Results

Given an image in JPEG format, our task is to detect if the image has been double compressed. To this end, the histograms of the DCT coefficients are computed. If these histograms contain periodic patterns, then the image is very likely to have been double compressed. Shown in Fig. 5 are the DCT coefficients and their histograms for an image that has been single JPEG compressed with qualities 75 (Fig. 5(a)) and 85 (Fig. 5(c)), and double JPEG compressed with qualities 85 followed by 75 (Fig. 5(b)), and 75 followed by 85 (Fig. 5(d)). The DCT coefficients are shown as images (auto-scaled to fill the full intensity range) where each pixel corresponds to a 8×8 block of the JPEG compressed image, and its intensity represents the coefficient value. These coefficients correspond to DCT frequencies $(1, 1)$ (the DC component) and $(2, 2)$. Note the presence of periodic artifacts in the histograms of the DCT coefficients of the double compressed images (Fig. 5(b) and 5(d)). Note also that these types of artifacts are not present in single compressed images (Fig. 5(a) and 5(c)). These periodic artifacts are particularly visible in the Fourier domain as strong peaks in the mid and high frequencies, Fig. 5(e).

The periodic patterns introduced by double JPEG compression depend on the quality parameters. As a result, it is possible to detect not only if an image has been double compressed, but also the compression qualities that have been used. The second parameter can be found from the quantization table stored in the JPEG file. The first parameter can be inferred from the location of the frequency peaks in the Fourier transforms of the DCT coefficient histograms.

4 Luminance Non-linearities

In order to enhance the perceptual quality of digital images, imaging devices often introduce some form of luminance non-linearity. The parameters of this non-linearity are usually dynamically chosen and depend on the camera and scene dynamics — these parameters are, however, typically held constant within an image. The presence of several distinct non-linearities in an image is a sign of possible tampering. For example, imagine a scenario where two images are spliced together. If the images were taken with different cameras or in different lightning conditions, then it is likely that different non-linearities are present in the composite image. It is also possible that local non-linearities are applied in the composite image in order to create a convincing luminance match.

We have previously proposed a technique to estimate parametric models of geometric and luminance non-linearities from digital images [6, 7]. This technique exploits the fact that a non-linear transformation introduces specific correlations in the Fourier domain. These correlations can be detected and estimated using tools from polyspectral analysis. This same technique can be employed to detect if an image contains multiple non-linearities. We describe below how luminance non-linearities introduce specific correlations, and how these correlations can be estimated.

4.1 Non-linearities and Correlations

Pointwise non-linear transformations introduce specific correlations in the frequency domain. To understand the form of these correlations, consider a one-dimensional discrete signal composed of a sum of two sinusoids with different phases and amplitudes: $x[t] = a_1 \cos(\omega_1 t + \phi_1) + a_2 \cos(\omega_2 t + \phi_2)$. Consider also a generic non-linear function $g(\cdot)$ and its Taylor series expansion where the various scalar constants and terms of degree higher than two are ignored: $g(u) \approx u + u.^2$. The non-linearly transformed signal takes the form:

$$\begin{aligned} g(x[t]) = &-0.5(a_1^2 + a_2^2) + a_1 \cos(\omega_1 t + \phi_1) + a_2 \cos(\omega_2 t + \phi_2) + 0.5a_1^2 \cos(2\omega_1 t + 2\phi_1) + \\ &0.5a_2^2 \cos(2\omega_2 t + 2\phi_2) + a_1 a_2 \cos((\omega_1 + \omega_2)t + (\phi_1 + \phi_2)) + a_1 a_2 \cos((\omega_1 - \omega_2)t + (\phi_1 - \phi_2)). \end{aligned} \quad (19)$$

Note that the non-linear transform introduced several new harmonics at frequencies $2\omega_1$, $2\omega_2$, $\omega_1 + \omega_2$, and $\omega_1 - \omega_2$. Note also that the phases of these new harmonics are correlated to the phases of the original ones. For example, the phase of harmonic $(\omega_1 + \omega_2)$ is equal to the sum of the phases of ω_1 and ω_2, and the phase of harmonic $2\omega_1$ is the double of the phase of harmonic ω_1. These type of correlations generalize to any type of underlying signal and pointwise non-linearity.

These phase correlations can be detected and estimated using tools from polyspectral analysis. Let $X(\omega)$ denote the Fourier transform of $x[t]$: $X(\omega) = \sum_{t=-\infty}^{\infty} x[t] e^{-it\omega}$. The power spectrum is a commonly employed

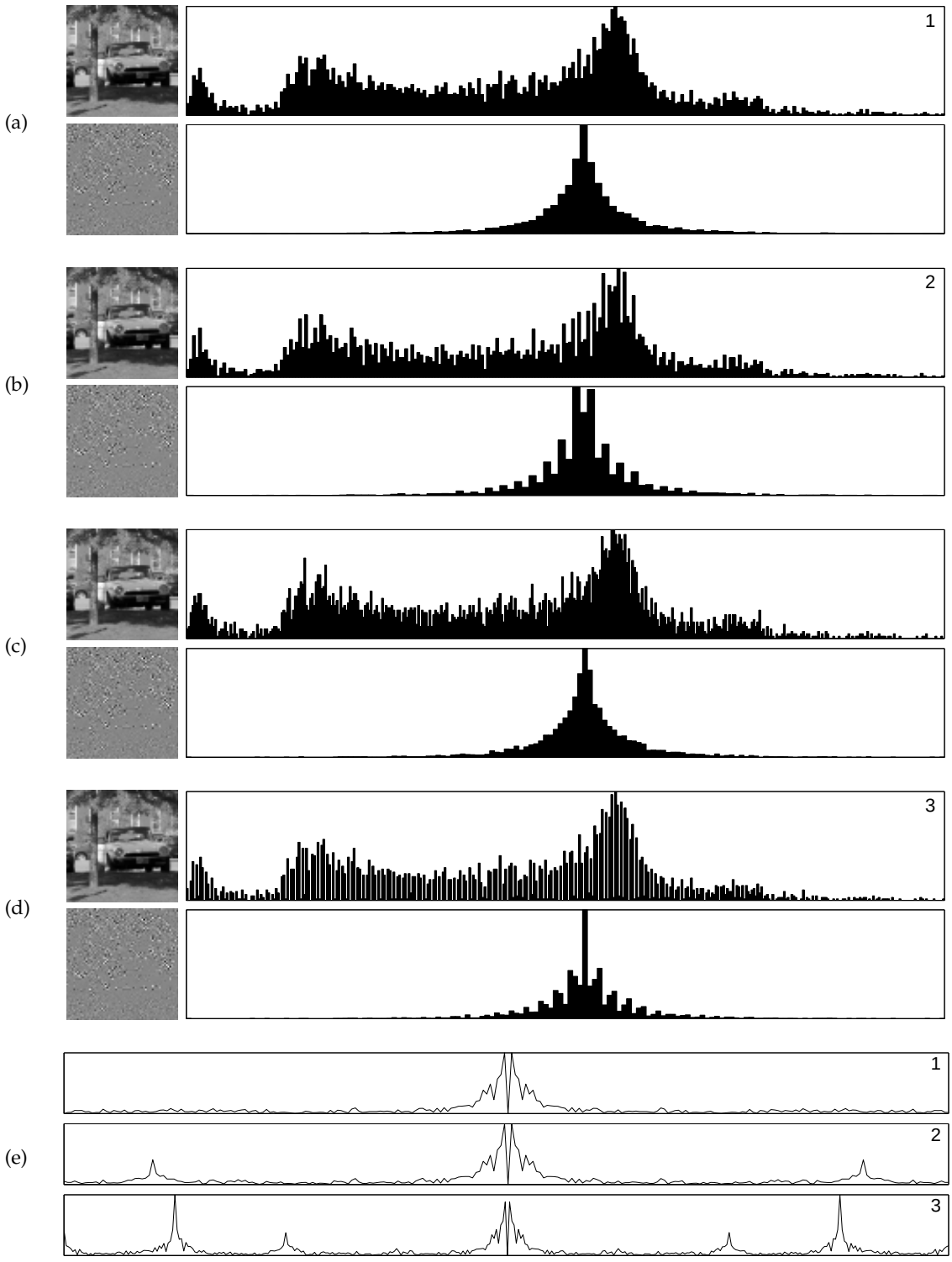

Fig. 5: Shown in the top four panels are DCT coefficients for two frequencies ($(1,1)$ and $(2,2)$), and their histograms for single and double compressed JPEG images: (a) single JPEG compression with quality 75, (b) double JPEG compression with quality 85 followed by 75, (c) single JPEG compression with quality 85, (d) double JPEG compression with quality 75 followed by 85. Shown in panel (e) are the Fourier transforms of three zero-meaned histograms. Note the periodic artifacts introduced by double quantization (panels $2, 3$) reflected by the high frequency peaks in the Fourier transforms.

tool to estimate second order correlations: $P(\omega) = \mathcal{E}\{X(\omega)X^*(\omega)\}$, where $\mathcal{E}\{\cdot\}$ is the expected value operator, and * denotes the complex conjugate. However the power spectrum is blind to higher-order correlations of the kind introduced by pointwise non-linearities. These correlations can be detected and estimated using higher-order spectra (see [20] for a thorough review). For example, the bispectrum can be employed to estimate third-order correlations: $B(\omega_1, \omega_2) = \mathcal{E}\{X(\omega_1)X(\omega_2)X^*(\omega_1 + \omega_2)\}$.

It can be seen intuitively that the bispectrum reveals correlations between harmonically related frequencies, such as $[\omega_1, \omega_1, 2\omega_1]$, $[\omega_2, \omega_2, 2\omega_2]$, $[\omega_1, \omega_2, \omega_1 + \omega_2]$, and $[\omega_1, \omega_2, \omega_1 - \omega_2]$. Under the assumption that the signal is ergodic, the bispectrum can be estimated as follows: divide $x[t]$ into N (possibly overlapping) segments, compute the Fourier transform of each segment k: $X_k(\omega)$, compute an average estimate of the bispectrum using the Fourier transform of individual segments $\hat{B}(\omega_1, \omega_2) = 1/N \sum_{k=1}^{N} X_k(\omega_1)X_k(\omega_2)X_k^*(\omega_1 + \omega_2)$. The bispectrum has the undesired property that its value at bi-frequency (ω_1, ω_2) depends on $P(\omega_1)$, $P(\omega_2)$, and $P(\omega_1 + \omega_2)$. For analysis purposes, it is useful to work with normalized quantities. To this end, we employ the bicoherence [13] (a normalized bispectrum), defined as:

$$b(\omega_1, \omega_2) = \frac{|B(\omega_1, \omega_2)|}{\left(\mathcal{E}\{|X(\omega_1)X(\omega_2)|^2\}\mathcal{E}\{|X(\omega_1 + \omega_2)|^2\}\right)^{1/2}}. \tag{20}$$

Note that the bicoherence is a real valued quantity, unlike the bispectrum. It is fairly straightforward to show using the Schwartz inequality [3] that the bicoherence is guaranteed to take values in $[0, 1]$. Just like the bispectrum, the bicoherence can be estimated as:

$$\hat{b}(\omega_1, \omega_2) = \frac{\frac{1}{K}\left|\sum_k X_k(\omega_1)X_k(\omega_2)X_k^*(\omega_1 + \omega_2)\right|}{\left(\left(\frac{1}{K}\sum_k |X_k(\omega_1)X_k(\omega_2)|^2\right)\left(\frac{1}{K}\sum_k |X_k(\omega_1 + \omega_2)|^2\right)\right)^{1/2}}. \tag{21}$$

This estimator will be used to measure third-order correlations.

4.2 Detecting Multiple Non-linearities

For simplicity, we assume that pointwise luminance non-linearities can be modeled with a one parameter family of functions of the form: $g(u) = u^\gamma$, where u denotes the intensity of a pixel normalized in the interval $[0, 1]$. We have previously shown that higher order correlations introduced by a non-linear transformation are proportional to the value of the parameter γ [6]. The following technique is used to blindly estimating the value of γ:

1. sample a range of inverse gamma values $1/\gamma$,
2. for each $1/\gamma$ in the selected range, apply the inverse function $g^{-1}(u) = u^{1/\gamma}$ to the signal, and compute the mean bicoherence $\sum_{\omega_1, \omega_2 = -\pi}^{\pi} \hat{b}(\omega_1, \omega_2)$.
3. select the inverse value $1/\gamma$ that minimizes the mean bicoherence.

Blindly estimating the value of γ from a gamma corrected image requires computing the bicoherence of a 2-D signal, a four-dimensional quantity. In order to avoid computational and memory requirements, the analysis will be restricted to horizontal and vertical scan lines of an image. This is reasonable since luminance non-linearities are usually pointwise transformations, and the type of correlations introduced in 1-D are similar to those in 2-D. The technique to estimate γ from an image is based on the one used for 1-D signals, as described above.

Shown in the top portion of Fig. 6 is a natural image (1200×1600 pixels in size) and the same image whose upper half has been gamma corrected with $\gamma = 1.8$. The bottom portion shows the estimated gamma values from horizontal scan lines of the unadulterated image (black dots) and the gamma corrected image (white dots). Notice that the values of the gamma estimates from scan lines that span the upper half of the tampered image are generally inconsistent with the lower half.

[3] Given two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, the Schwartz inequality states: $\|\boldsymbol{x}\|\|\boldsymbol{y}\| \geq |\boldsymbol{x} \cdot \boldsymbol{y}|$, where $\|\cdot\|$ denotes vector norm, and $\cdot$ denotes scalar product.

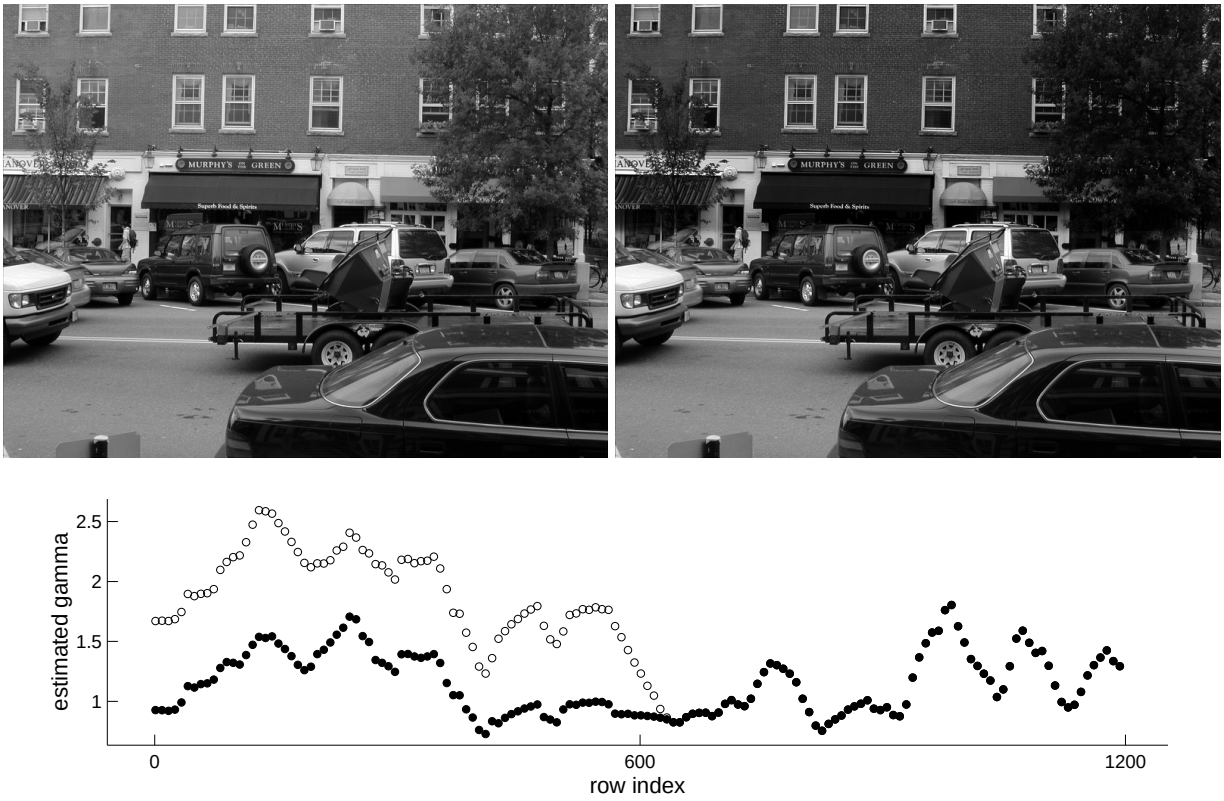

Fig. 6: Top panel: a natural image (left), and the same image whose top portion was gamma corrected with $\gamma = 1.8$ (right). The images are 1200×1600 pixels in size. Bottom panel: Estimated gamma values from horizontal scan lines, where the black dots correspond to estimates from the unadulterated image, and the white dots correspond to estimates from the image whose upper half has been gamma corrected. Each data point corresponds to a running average over 60 scan lines.

5 Signal to Noise Ratio

Digital images have an inherent amount of noise introduced either by the imaging process or digital compression. The amount of noise is typically uniform across the entire image. If two images with different noise levels are spliced together, or if small amounts of noise are locally added to conceal traces of tampering, then variations in the signal to noise ratio (SNR) across the image can be used as evidence of tampering. Measuring the SNR is non-trivial in the absence of the original signal. Several *blind* SNR estimators have, however, been proposed [22]. We first describe one such estimator, M_2M_4 [19], and then show its effectiveness in locally measuring noise variance (so as to be invariant to the underlying signal strength, we analyze the noise variance instead of the ratio of signal to noise variances).

We begin by assuming an additive noise model: $y[t] = x[t] + w[t]$, where $x[t]$ is the uncorrupted signal with variance S and $w[t]$ is the noise with variance N. Denote the second and forth moments of the corrupted signal as $M_2 = \mathcal{E}\left\{y^2[t]\right\}$ and $M_4 = \mathcal{E}\left\{y^4[t]\right\}$, where $\mathcal{E}\left\{\cdot\right\}$ is the expected value operator. Assuming that the signal and noise are independent and zero-mean, it can be shown [22] that:

$$M_2 = S + N \qquad \text{and} \qquad M_4 = k_x S^2 + 6SN + k_w N^2, \tag{22}$$

where $k_x = \mathcal{E}\left\{x^4[t]\right\} / (\mathcal{E}\left\{x^2[t]\right\})^2$ and $k_w = \mathcal{E}\left\{w^4[t]\right\} / (\mathcal{E}\left\{w^2[t]\right\})^2$ are the kurtoses of the original signal and noise. Solving Equation (22) for S and N yields:

$$S = \frac{M_2(k_w - 3) \pm \sqrt{(9 - k_x k_w)M_2^2 + M_4(k_x + k_w - 6)}}{k_x + k_y - 6} \qquad \text{and} \qquad N = M_2 - S. \tag{23}$$

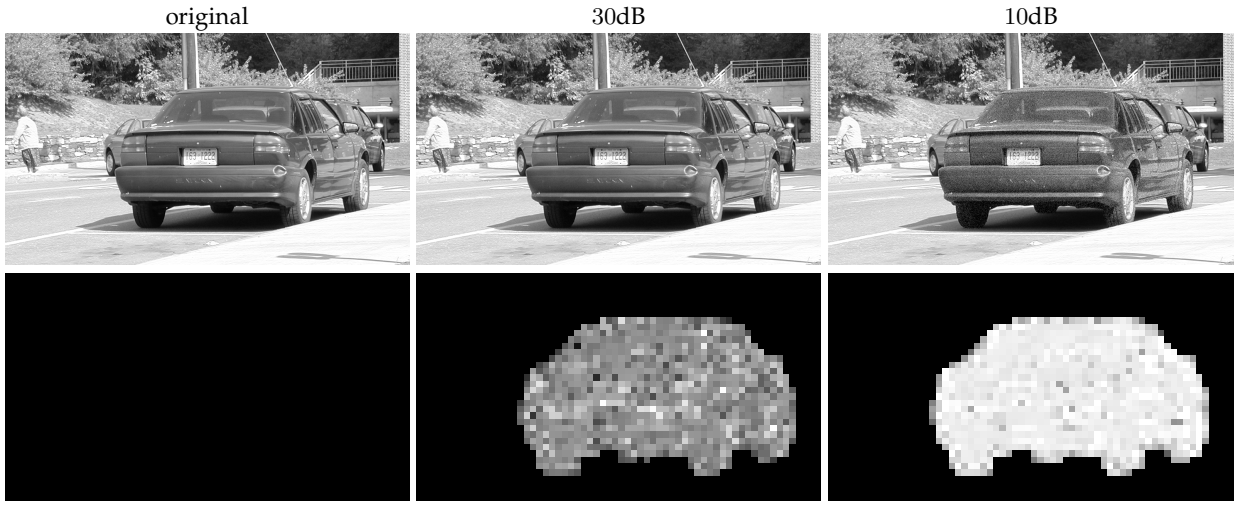

Fig. 7: Shown on the top row is an original image and this image with noise added locally to the car. Shown on the bottom row are the locally estimated noise variances (on the same log scale).

Note that this estimator assumes a known kurtosis for the original signal and the noise, k_x and k_w. In general these quantities may not be known. In the results presented below, we assume that they are known. In the future, the kurtosis of the original signal can be estimated from a region of an image that is believed to be authentic, and the kurtosis of the noise can be estimated by, for example, assuming Gaussian noise ($k_w = 3$), or modeling the noise statistics of JPEG compression.

Shown in the top row of Fig. 7 is an original image, and this image with additive white Gaussian noise with SNRs of 30dB (N=0.08×10^{-3}) and 10dB (N=7.62×10^{-3}) added locally to only the car. Shown in the bottom row of this figure are the estimated noise variances from overlapping (by 32 pixels) 64×64 blocks. The average estimated noise variances, for the blocks overlapping the car, are 0.25×10^{-3} and 7.20×10^{-3}. Notice that the estimator is easily able to detect different noise levels in the image.

6 Discussion

We have described a set of statistical tools for detecting traces of digital tampering in the absence of any digital watermark or signature. We have quantified the nature of statistical correlations that result from specific forms of digital tampering, and have devised detection schemes to reveal these correlations. We are currently developing other tools that, in the same spirit of those presented here, reveal statistical correlations that result from a variety of different manipulations that are typically necessary to create a convincing digital forgery. We are also analyzing the sensitivity and robustness to counter-attack of each of the schemes outlined in this paper.

There is little doubt that counter-measures will be developed to foil each of the detection schemes outlined in this paper. Our hope, however, is that as more authentication tools are developed it will become increasingly more difficult to create convincing digital forgeries. In addition, as the suite of detection tools expands we believe that it will become increasingly harder to simultaneously foil each of the detection schemes.

References

1. Digital compression and coding of continuous-tone still images, Part 1: Requirements and guidelines. ISO/IEC JTC1 Draft International Standard 10918-1, 1991.
2. S. Bhattacharjee and M. Kutter. Compression-tolerant image authentication. In *IEEE International Conference on Image Processing*, 1998.
3. I.J. Cox, M.L. Miller, and J.A. Bloom. *Digital Watermarking*. Morgan Kaufmann Publishers, 2002.
4. S.A. Craver, M. Wu, B. Liu, A. Stubblefield, B. Swartzlander, and D.S. Wallach. Reading between the lines: Lessons from the SDMI challenge. In *10th USENIX Security Symposium*, 2001.
5. A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum lilelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 99(1):1–38, 1977.
6. H. Farid. Blind inverse gamma correction. *IEEE Transactions on Image Processing*, 10(10):1428–1433, 2001.
7. H. Farid and A.C. Popescu. Blind removal of lens distortions. *Journal of the Optical Society of America*, 18(9):2072–2078, 2001.
8. J. Fridrich and M. Goljan. Images with self-correcting capabilities. In *IEEE International Conference on Image Processing*, 1999.
9. J. Fridrich, M. Goljan, and M. Du. Invertible authentication. In *SPIE, Security and Watermarking of Multimedia Contents*, 2001.
10. G.L. Friedman. The trustworthy camera: Restoring credibility to the photographic image. *IEEE Transactions on Consumer Electronics*, 39(4):905–910, 1993.
11. C.W. Honsinger, P.Jones, M.Rabbani, and J.C. Stoffel. Lossless recovery of an original image containing embedded data. U.S. Patent Application, Docket No. 77102/E-D, 1999.
12. S. Katzenbeisser and F.A.P. Petitcolas. *Information Techniques for Steganography and Digital Watermarking*. Artec House, 2000.
13. Y.C. Kim and E.J. Powers. Digital bispectral analysis and its applications to nonlinear wave interactions. *IEEE Transactions of Plasma Science*, PS 7(2), 1979.
14. D. Kundur and D. Hatzinakos. Digital watermarking for tell-tale tamper proofing and authentication. *Proceedings of the IEEE*, 87(7):1167–1180, 1999.
15. C.-Y. Lin and S.-F. Chang. A robust image authentication algorithm surviving JPEG lossy compression. In *SPIE Storage and Retrieval of Image/Video Databases*, 1998.
16. E.T. Lin, C.I Podilchuk, and E.J. Delp. Detection of image alterations using semi-fragile watermarks. In *SPIE International Conference on Security and Watermarking of Multimedia Contents II*, 2000.
17. J. Lukas and J. Fridrich. Estimation of primary quantization matrix in double compressed JPEG images. In *Digital Forensic Research Workshop*, Cleveland, Ohio, August 2003.
18. B.M. Macq and J.-J. Quisquater. Cryptology for digital TV broadcasting. *Proceedings of the IEEE*, 83(6):944–957, 1995.
19. R. Matzner. An SNR estimation algorithm for complex baseband signals using higher order statistics. *Facta Universitatis (Nis)*, 6(1):41–52, 1993.
20. J.M. Mendel. Tutorial on higher-order statistics (spectra) in signal processing and system theory: Theoretical results and some applications. *Proceedings of the IEEE*, 79(3):278–305, 1991.
21. A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, 1989.
22. D.R. Pauluzzi and N.C. Beaulieu. A comparison of SNR estimation techniques for the AWGN channel. *IEEE Transactions on Communications*, 48(10):1681–1691, 2000.
23. C. Rey and J.-L. Dugelay. Blind detection of malicious alterations on still images using robust watermarks. In *IEE Seminar: Secure Images and Image Authentication*, 2000.
24. M. Schneider and S.-F. Chang. A robust content-based digital signature for image authentication. In *IEEE International Conference on Image Processing*, 1996.
25. D. Storck. A new approach to integrity of digital images. In *IFIP Conference on Mobile Communication*, pages 309–316, 1996.
26. G.K. Wallace. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 34(4):30–44, 1991.
27. M. Yeung and F. Mintzer. An invisible watermarking technique for image verification. In *IEEE International Conference on Image Processing*, 1997.
28. G.-J. Yu, C.-S. Lu, H.-Y.M. Liao, and J.-P. Sheu. Mean quantization blind watermarking for image authentication. In *IEEE International Conference on Image Processing*, 2000.