Exposing Digital Forgeries by Detecting Traces of Re-sampling

Alin C. Popescu and Hany Farid†

Abstract

The unique stature of photographs as a definitive recording of events is being diminished due, in part, to the ease with which digital images can be manipulated and altered. Although good forgeries may leave no visual clues of having been tampered with, they may, nevertheless, alter the underlying statistics of an image. For example, we describe how re-sampling (e.g., scaling or rotating) introduces specific statistical correlations, and describe how these correlations can be automatically detected in any portion of an image. This technique works in the absence of any digital watermark or signature. We show the efficacy of this approach on uncompressed TIFF images, and JPEG and GIF images with minimal compression. We expect this technique to be among the first of many tools that will be needed to expose digital forgeries.

I. Introduction

With the advent of low-cost and high-resolution digital cameras, and sophisticated editing software, digital images can be easily manipulated and altered. Digital forgeries, often leaving no visual clues of having been tampered with, can be indistinguishable from authentic photographs. As a result, photographs no longer hold the unique stature as a definitive recording of events. Of particular concern is how the judicial system and news media will contend with this issue. For example, in March of 2003 the *Los Angeles Times* published, on its front page, a dramatic photograph of a soldier directing an Iraqi citizen to take cover. The photograph, however, was a fake - it was digitally created by splicing together two photographs [1]. This and similar incidents naturally lead us to wonder how many of the images that we see every day have been digitally doctored.

Digital watermarking has been proposed as a means by which an image can be authenticated (see, for example, [1], [2] for general surveys). Within this broad area, several authentication schemes have been proposed: embedded signatures [3], [4], [5], [6], [7], eraseable fragile watermarks [8], [9], semi-fragile watermarks [10], [11], [12], [13], robust tell-tale watermarks [14], [15], [12], [16], [17], and self-embedding watermarks [18]. 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 or

A. C. Popescu is with the Computer Science Department at Dartmouth College.

Corresponding author: H. Farid, 6211 Sudikoff Lab, Computer Science Department, 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, a National Science Foundation CAREER Award (IIS-99-83806), a Department of Justice Grant (2000-DT-CS-K001), and a departmental National Science Foundation Infrastructure Grant (EIA-98-02068).

[1]The fake was discovered when an editor at *The Hartford Courant* noticed that civilians in the background appeared twice in the photo.

image header. With the assumption that digital tampering will alter a watermark (or signature), 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 digital watermark cannot be easily removed and reinserted - it is not yet clear whether this is a reasonable assumption (e.g., [19]).

In contrast to these approaches, we describe a technique for detecting traces of digital tampering in the complete absence of any form of digital watermark or signature. This approach works 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. For example, consider 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 re-size, rotate, or stretch portions of the images. This process requires re-sampling the original image onto a new sampling lattice. Although this re-sampling is often imperceptible, it introduces specific correlations into the image, which when detected can be used as evidence of digital tampering. We describe the form of these correlations, and how they can be automatically detected in any portion of an image. We show the general effectiveness of this technique and analyze its sensitivity and robustness to counter-attacks.

II. Re-sampling

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.

A. Re-sampling Signals

Consider a 1-D discretely-sampled signal $x[t]$ with m samples, Fig. 1(a). The number of samples in this signal can be increased or decreased by a factor p/q to n samples in three steps [20]:

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, Fig. 1(b).
2) interpolate: convolve $x_u[t]$ with a low-pass filter: $x_i[t] = x_u[t] \star h[t]$, Fig. 1(c).
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 $y[t] \equiv x_d[t]$, Fig. 1(d).

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, $\vec{x}$ and $\vec{y}$, respectively, re-sampling

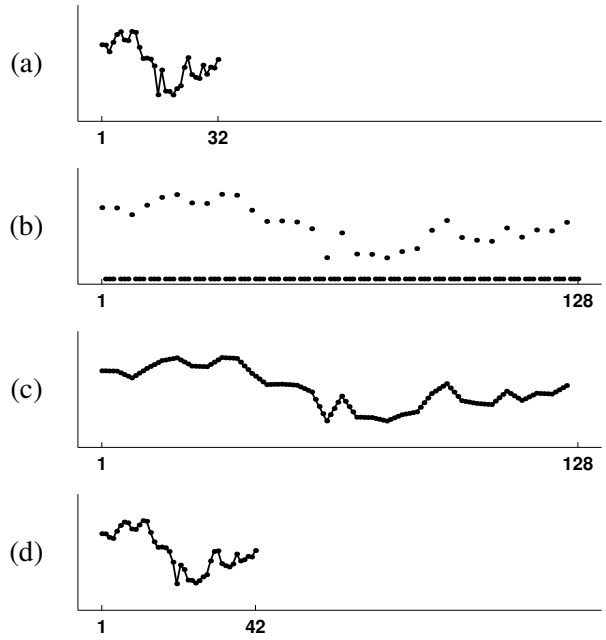


Fig. 1. Re-sampling a signal by a factor of $4/3$: shown are (a) the original signal; (b) the up-sampled signal; (c) the interpolated signal; and (d) the final re-sampled signal.

takes the form:

$$\vec{y} \;=\; A_{p/q}\vec{x}, \tag{1}$$

where the $n \times m$ matrix $A_{p/q}$ embodies the entire re-sampling process. For example, the matrix for up-sampling by a factor of $4/3$ using linear interpolation (Fig. 1) has the form:

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

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. In this case, the re-sampling matrix takes the form:

$$A_{2/1} = \begin{bmatrix} 1 & 0 & 0 & \\ 0.5 & 0.5 & 0 & \\ 0 & 1 & 0 & \\ 0 & 0.5 & 0.5 & \\ 0 & 0 & 1 & \\ & & & \ddots \end{bmatrix}. \quad (3)$$

Here, the odd samples of the re-sampled signal $\vec{y}$ take on the values of the original signal $\vec{x}$, 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:

$$y_{2i} = 0.5x_i + 0.5x_{i+1}, \quad (4)$$

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:

$$y_{2i} = 0.5y_{2i-1} + 0.5y_{2i+1}. \quad (5)$$

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:

$$y_i \stackrel{?}{=} \sum_{k=-N}^{N} \alpha_k y_{i+k}, \quad (6)$$

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:

$$y_i - \sum_{k=-N}^{N} \alpha_k y_{i+k} = 0 \quad (7)$$

$$(\vec{a}_i \cdot \vec{x}) - \sum_{k=-N}^{N} \alpha_k (\vec{a}_{i+k} \cdot \vec{x}) = 0 \quad (8)$$

$$\left(\vec{a}_i - \sum_{k=-N}^{N} \alpha_k \vec{a}_{i+k} \right) \cdot \vec{x} = 0, \quad (9)$$

where $\vec{a}_i$ is the i^{th} row of the re-sampling matrix $A_{p/q}$, and $\vec{x}$ 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, $\vec{a}_i$, is equal to a linear combination of the neighboring rows, $\sum_{k=-N}^{N} \alpha_k \vec{a}_{i+k}$. For example, in the case of

up-sampling by a factor of two, Equation (3), 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 (9) to be satisfied when the difference on the left-hand side of the equation is orthogonal to the original signal $\vec{x}$. While this may occur on occasion, these correlations are unlikely to be periodic.

B. 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. Consider again the re-sampling matrix of Equation (2). 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, $\vec{\alpha}$, that satisfy: $\vec{a}_i = \sum_{k=-N}^{N} \alpha_k \vec{a}_{i+k}$, Equation (9), where $\vec{a}_i$ 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, $\vec{\alpha}$, then it is straight-forward to determine which samples satisfy $y_i = \sum_{k=-N}^{N} \alpha_k y_{i+k}$, Equation (7).

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) [21] 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 can be obtained using Bayes' rule:

$$\Pr\{y_i \in M_1 \mid y_i\} = \frac{\Pr\{y_i \mid y_i \in M_1\} \Pr\{y_i \in M_1\}}{\sum_{k=1}^{2} \Pr\{y_i \mid y_i \in M_k\} \Pr\{y_i \in M_k\}}, \tag{10}$$

where the priors $\Pr\{y_i \in M_1\}$ and $\Pr\{y_i \in M_2\}$ are assumed to be equal to $1/2$. We also assume that

$$\Pr\{y_i | y_i \in M_1\} = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-\left(y_i - \sum_{k=-N}^{N} \alpha_k y_{i+k}\right)^2}{2\sigma^2}\right], \tag{11}$$

and that $\Pr\{y_i | y_i \in M_2\}$ is uniformly distributed over the range of possible values of the signal $\vec{y}$. The variance, σ, of the Gaussian distribution in Equation (11) is estimated in the M-step (see Appendix A). Note that the E-step

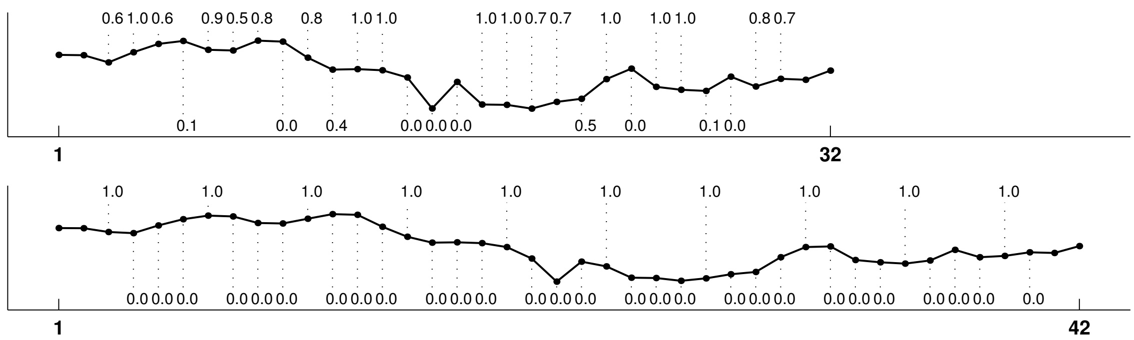


Fig. 2. A signal with 32 samples (top) and this signal re-sampled by a factor of $4/3$ (bottom). Each sample is annotated with its probability of being correlated to its neighbors. Note that for the up-sampled signal these probabilities are periodic, while for the original signal they are not.

requires an estimate of $\vec{\alpha}$, which on the first iteration is chosen randomly. In the M-step, a new estimate of $\vec{\alpha}$ is computed using weighted least-squares, that is, minimizing the following quadratic error function:

$$E(\vec{\alpha}) \quad = \quad \sum_i w(i) \left(y_i - \sum_{k=-N}^{N} \alpha_k y_{i+k} \right)^2, \tag{12}$$

where the weights $w(i) \equiv \Pr\{y_i \in M_1 | y_i\}$, Equation (10), and $\alpha_0 = 0$. This error function is minimized by computing the gradient with respect to $\vec{\alpha}$, setting the result equal to zero, and solving for $\vec{\alpha}$, yielding:

$$\vec{\alpha} \quad = \quad (Y^T W Y)^{-1} Y^T W \vec{y}, \tag{13}$$

where the matrix Y is:

$$Y = \begin{bmatrix} y_1 & \cdots & y_N & y_{N+2} & \cdots & y_{2N+1} \\ y_2 & \cdots & y_{N+1} & y_{N+3} & \cdots & y_{2N+2} \\ \vdots & & \vdots & \vdots & & \vdots \\ y_i & \cdots & y_{N+i-1} & y_{N+i+1} & \cdots & y_{2N+i} \\ \vdots & & \vdots & \vdots & & \vdots \end{bmatrix}, \tag{14}$$

and W is a diagonal weighting matrix with $w(i)$ along the diagonal. The E-step and the M-step are iteratively executed until a stable estimate of $\vec{\alpha}$ is achieved (see Appendix A for more details).

Shown in Fig. 2 are the results of running EM on the original and re-sampled signals of Fig. 1. 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 later case, the periodic pattern is obvious, where only every 4^{th} sample has probability 1, as would be expected from an up-sampling by a factor of $4/3$, Equation (2). 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. It is not possible, however, to uniquely determine the specific amount of re-sampling. The reason is that although periodic patterns

may be unique for a set of re-sampling parameters, there are parameters that will produce similar patterns. For example, re-sampling by a factor of $3/4$ and by a factor of $5/4$ will produce indistinguishable periodic patterns. As a result, we can only estimate the amount of re-sampling within this ambiguity. Since we are primarily concerned with detecting traces of re-sampling, and not necessarily the amount of re-sampling, this limitation is not critical.

There is 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). The re-sampling matrix, in this case, is given by:

$$A_{1/2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 1 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 1 & \\ & & & & & \ddots \end{bmatrix}. \tag{15}$$

Notice that no row can be written as a linear combination of the neighboring rows - in this case, re-sampling is not detectable. More generally, the detectability of any re-sampling can be determined by generating the re-sampling matrix and determing whether any rows can be expressed as a linear combination of their neighboring rows - a simple empirical algorithm is described in Section III-A.

C. 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. Though we will only show this for up- and down-sampling, the same is true for an arbitrary affine transform (and more generally for any non-linear geometric transformation).

Consider, for example, the simple case of up-sampling by a factor of two. Shown in Fig. 3 is, from left to right, a portion of an original 2-D sampling lattice, the same lattice up-sampled by a factor of two, and a subset of the pixels of the re-sampled image. Assuming linear interpolation, these pixels are given by:

$$\begin{aligned} y_2 &= 0.5y_1 + 0.5y_3 \\ y_4 &= 0.5y_1 + 0.5y_7 \\ y_5 &= 0.25y_1 + 0.25y_3 + 0.25y_7 + 0.25y_9, \end{aligned} \tag{16}$$

where $y_1 = x_1$, $y_3 = x_2$, $y_7 = x_3$, $y_9 = x_4$. Note that all the pixels of the re-sampled image in the odd rows and even columns (e.g., y_2) will all be the same linear combination of their two horizontal neighbors. Similarly, the pixels of the re-sampled image in the even rows and odd columns (e.g., y_4) will all be the same linear combination

x_1	x_2
x_3	x_4

x_1	0	x_2
0	0	0
x_3	0	x_4

y_1	y_2	y_3
y_4	y_5	
y_7		y_9

Fig. 3. Shown from left to right are: a portion of the 2D lattice of an image, the same lattice up-sampled by a factor of two, and a portion of the lattice of the re-sampled image.

of their two 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 these periodic correlations with 1-D signals, the same approach can be used with 2-D images.

III. 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 II-B was used to estimate probability maps that embody the correlation between each pixel and its neighbors. The EM parameters were fixed throughout at $N = 2$, $\sigma_0 = 0.0075$, and $N_h = 3$ [2] (see Appendix A). Shown in Figs. 4-6 are several examples of the periodic patterns that emerged due to re-sampling. In the top row of each figure are (from left to right) the original 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 is the same image uniformly re-sampled at different rates. For the re-sampled images, note the periodic nature of their probability maps and the localized peaks in their corresponding Fourier transforms. Shown in Fig. 7 are examples of the periodic patterns that emerge from four different affine transforms. Shown in Fig. 8 are the results from applying consecutive re-samplings. Specifically, the image in the top row was first upsampled by 15% and then this up-sampled image was rotated by 5°. The same operations were performed in reverse order on the image in the bottom row. Note that while the images are perceptually indistinguishable, the periodic patterns that emerge are quite distinct, with the last re-sampling operation dominating the pattern. Note, however, that the corresponding Fourier transforms contain several sets of peaks corresponding to both re-sampling operations. As with a single re-sampling, consecutive re-samplings are easily detected.

[2]The blurring of the residual error with a binomial filter of width N_h is not critical, but merely accelerates the convergence of EM.

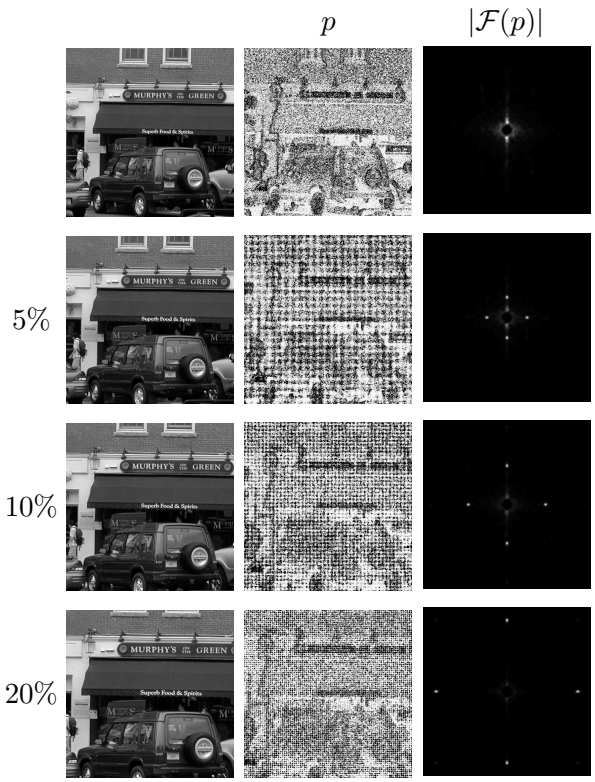


Fig. 4. Shown in the top row is the original image, and shown below is the same image up-sampled by varying amounts. Shown in the middle column are the estimated probability maps (p) that embody the spatial correlations in the image. The Fourier transform of each map is shown in the right-most column. Note that only the re-sampled images yield periodic maps.

Shown in Figs. 9-10 are examples of our detection algorithm applied to images where only a portion of the image was re-sampled. Regions in each image were subjected to a range of stretching, rotation, shearing, etc. (these manipulations were done in Adobe Photoshop using bi-cubic interpolation). Shown in each figure is the original photograph, the forgery, and the estimated probability map. Note that in each case, the re-sampled region is clearly detected - while the periodic patterns are not particularly visible in the spatial domain at the reduced scale, the well localized peaks in the Fourier domain clearly reveal their presence (for display purposes, each Fourier transform was independently auto-scaled to fill the full intensity range and high-pass filtered to suppress the lowest frequencies). Note also that in Fig. 9 the white sheet of paper on top of the trunk has strong activation in the probability map - when seen in the Fourier domain, however, it is clear that this region is not periodic, but rather is uniform, and thus not representative of a re-sampled region.

A. Sensitivity and Robustness

From a digital forensic perspective it is important to quantify the robustness and sensitivity of our detection algorithm. To this end, it is first necessary to devise a quantitative measure of the extent of periodicity found in the estimated probability maps. To do so, we compare the estimated probability map with a set of synthetically generated probability maps that contain periodic patterns similar to those that emerge from re-sampled images.

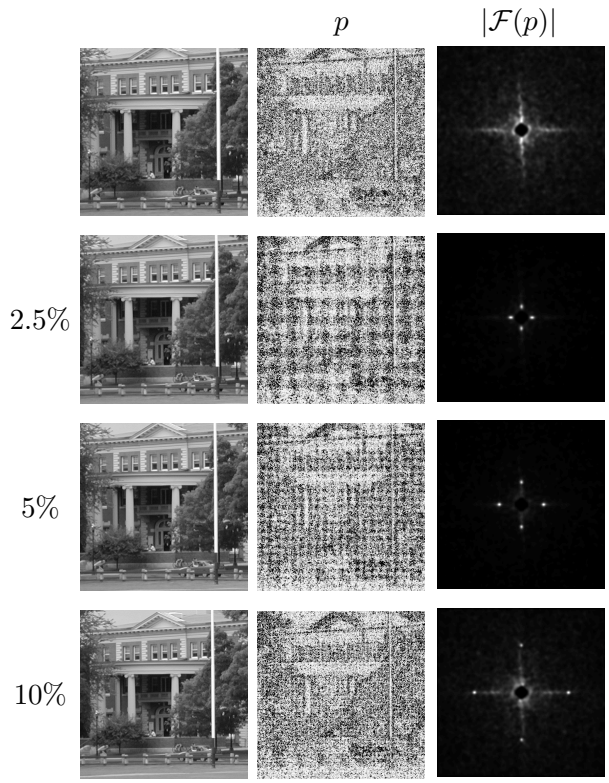


Fig. 5. Shown in the top row is the original image, and shown below is the same image down-sampled by varying amounts. Shown in the middle column are the estimated probability maps (p) that embody the spatial correlations in the image. The Fourier transform of each map is shown in the right-most column. Note that only the re-sampled images yield periodic maps.

Given a set of re-sampling parameters and interpolation method, a synthetic map is generated based on the periodicity of the re-sampling matrix. Note, however, that there are several possible periodic patterns that may emerge in a re-sampled image. For example, in the case of up-sampling by a factor of two using linear interpolation, Equation (16), the coefficients $\vec{\alpha}$ estimated by the EM algorithm (with a 3×3 neighborhood) are expected to be one of the following:

$$\vec{\alpha}_1 = \begin{bmatrix} 0 & 0.5 & 0 \\ 0 & 0 & 0 \\ 0 & 0.5 & 0 \end{bmatrix} \vec{\alpha}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0.5 & 0 & 0.5 \\ 0 & 0 & 0 \end{bmatrix} \vec{\alpha}_3 = \begin{bmatrix} 0.25 & 0 & 0.25 \\ 0 & 0 & 0 \\ 0.25 & 0 & 0.25 \end{bmatrix}. \tag{17}$$

We have observed that EM will return one of these estimates only when the initial value of $\vec{\alpha}$ is close to one of the above three values, the neighborhood size is 3, and the initial variance of the conditional probability (σ in Equation (11)) is relatively small. In general, however, this fine tuning of the starting conditions is not practical. To be broadly applicable, we randomly choose an initial value for $\vec{\alpha}$, and set the neighborhood size and initial value of σ to values that afford convergence for a broad range of re-sampling parameters. Under these conditions, we have found that for specific re-sampling parameters and interpolation method, the EM algorithm typically converges to a unique set of linear coefficients. In the above example of up-sampling by a factor of two the EM algorithm

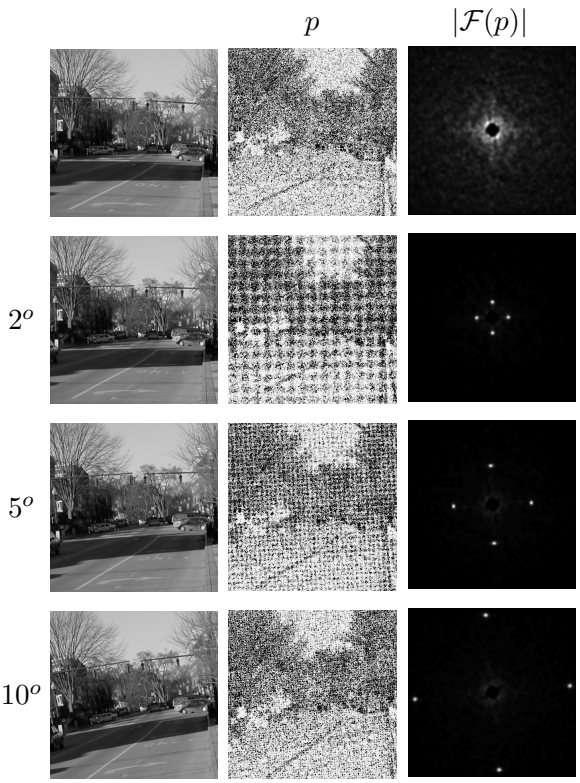


Fig. 6. Shown in the top row is the original image, and shown below is the same image rotated by varying amounts. Shown in the middle column are the estimated probability maps (p) that embody the spatial correlations in the image. The Fourier transform of each map is shown in the right-most column. Note that only the re-sampled images yield periodic maps.

typically converges to:

$$\vec{\alpha} = \begin{bmatrix} -0.25 & 0.5 & -0.25 \\ 0.5 & 0 & 0.5 \\ -0.25 & 0.5 & -0.25 \end{bmatrix}. \tag{18}$$

Note that this solution is different than each of the solutions in Equation (17). Yet, the relationships in Equation (16) are still satisfied by this choice of coefficients. Since the EM algorithm typically converges to a unique set of linear coefficients, there is also a unique periodic pattern that emerges. It is possible to predict this pattern by analyzing the periodic patterns that emerge from a large set of images. In practice, however, this approach is computationally demanding, and therefore we employ a simpler method that was experimentally determined to generate similar periodic patterns. This method first warps a rectilinear integer lattice according to a specified set of re-sampling parameters. From this warped lattice, the synthetic map is generated by computing the minimum distance between a warped point and an integer sampling lattice. More specifically, let M denote a general affine transform which embodies a specific re-sampling. Let (x, y) denote the points on an integer lattice, and $(\tilde{x}, \tilde{y})$ denote the points of

Fig. 7. Shown are four images affine transformed by random amounts. Shown in the middle column are the estimated probability maps (p) that embody the spatial correlations in the image. The Fourier transform of each map is shown in the right-most column. Note that these images yield periodic maps.

a lattice obtained by warping the integer lattice (x, y) according to M:

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} \quad = \quad M \begin{bmatrix} x \\ y \end{bmatrix}. \tag{19}$$

The synthetic map, $s(x, y)$, corresponding to M is generated by computing the minimum distance between each point in the warped lattice $(\tilde{x}, \tilde{y})$ to a point in the integer lattice:

$$s(x, y) \quad = \quad \min_{x_0, y_0} \sqrt{(\tilde{x} - x_0)^2 + (\tilde{y} - y_0)^2}, \tag{20}$$

where x_0 and y_0 are integers, and $(\tilde{x}, \tilde{y})$ are functions of (x, y) as given in Equation (19). Synthetic maps generated using this method are similar to the experimentally determined probability maps, Fig. 11.

The similarity between an estimated probability map, $p(x, y)$, and a synthetic map, $s(x, y)$, is computed as follows:

1) The probability map p is Fourier transformed: $P(\omega_x, \omega_y) = \mathcal{F}(p(x, y) \cdot W(x, y))$, where the radial portion of the rotationally invariant window, $W(x, y)$, takes the form:

$$f(r) = \begin{cases} 1 & 0 \leq r < 3/4 \\ \frac{1}{2} + \frac{1}{2} \cos\left(\frac{\pi(r-3/4)}{\sqrt{2}-3/4}\right) & 3/4 \leq r \leq \sqrt{2}, \end{cases} \tag{21}$$

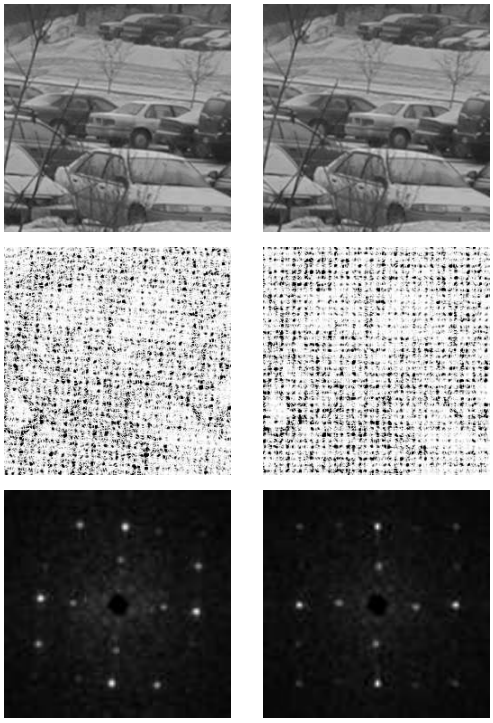


Fig. 8. Shown are two images that were consecutively re-sampled (top left: upsampled by 15% and then rotated by 5°; top right : rotated by 5° and then upsampled by 15%). Shown in the second row are the estimated probability maps that embody the spatial correlations in the image. The magnitude of the Fourier transform of each map is shown in the bottom column - note the multiple set of peaks that correspond to both the rotation and up-sampling.

where the radial axis is normalized between 0 and $\sqrt{2}$. Note that for notational convenience the spatial arguments on $p(\cdot)$ and $P(\cdot)$ will be dropped.

2) The Fourier transformed map P is then high-pass filtered to remove undesired low frequency noise: $P_H = P \cdot H$, where the radial portion of the rotationally invariant highpass filter, H, takes the form:

$$h(r) \quad = \quad \frac{1}{2} - \frac{1}{2}\cos\left(\frac{\pi r}{\sqrt{2}}\right), \quad 0 \leq r \leq \sqrt{2}. \tag{22}$$

3) The high-passed spectrum P_H is then normalized, gamma corrected in order to enhance frequency peaks, and then rescaled back to its original range:

$$P_G \quad = \quad \left(\frac{P_H}{\max(|P_H|)}\right)^4 \times \max(|P_H|). \tag{23}$$

4) The synthetic map s is simply Fourier transformed: $S = \mathcal{F}(s)$.

5) The measure of similarity between p and s is then given by:

$$M(p,s) = \sum_{\omega_x,\omega_y} |P_G(\omega_x,\omega_y)| \cdot |S(\omega_x,\omega_y)|, \tag{24}$$

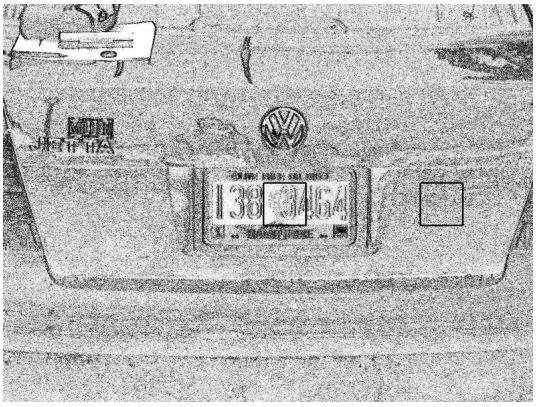
where $|\cdot|$ denotes absolute value (note that this similarity measure is phase insensitive).

A set of synthetic probability maps are first generated from a number of different re-sampling parameters. For a given probability map p, the most similar synthetic map, $s^\star$, is found through a brute-force search over the entire

original

forgery

probability map (p)



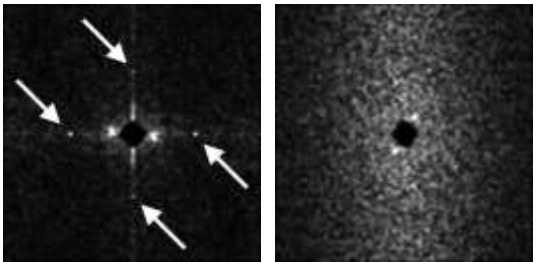


Fig. 9. Shown are the original image and a forgery. The forgery consists of splicing in a new license plate number. Shown below is the estimated probability map (p) of the forgery, and the magnitude of the Fourier transform ($\mathcal{F}(p)$) of a region in the license plate (left) and on the car trunk (right). The periodic pattern (spikes in $\mathcal{F}(p)$) in the license plate suggests that this region was re-sampled.

original

forgery

probability map (p)

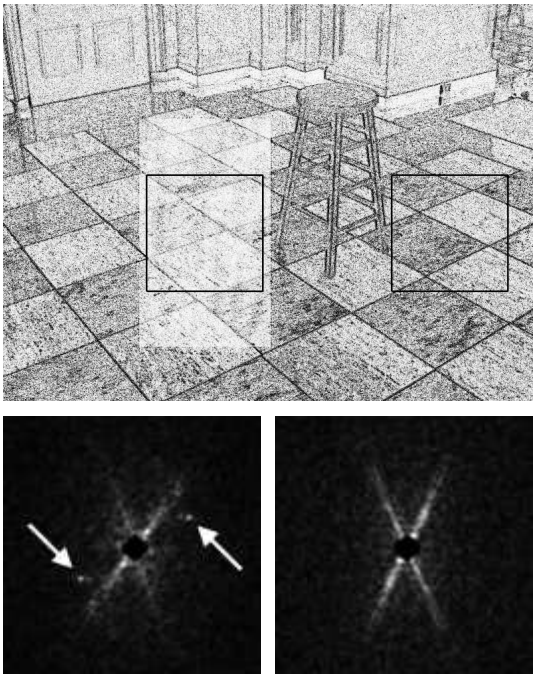


Fig. 10. Shown are the original image and a forgery. The forgery consists of removing a stool and splicing in a new floor taken from another image of the same room. Shown below is the estimated probability map (p) of the forgery, and the magnitude of the Fourier transform ($\mathcal{F}(p)$) of a region in the new floor (left) and on the original floor (right). The periodic pattern (spikes in $\mathcal{F}(p)$) in the new floor suggests that this region was re-sampled.

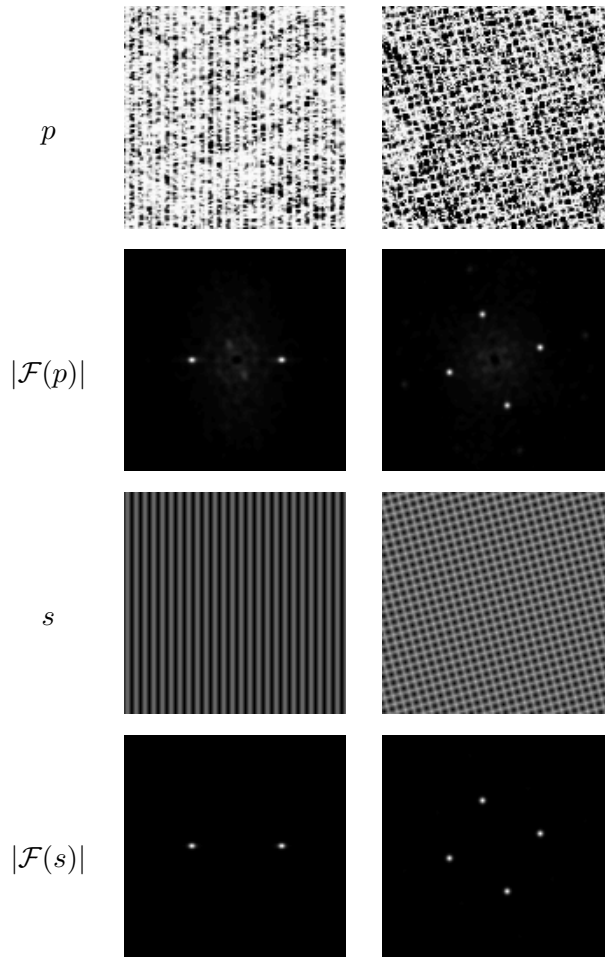


Fig. 11. Shown in the first two rows are estimated probability maps, p, from images that were re-sampled (affine transformed), and the magnitude of the Fourier transform of these maps. Note the strong periodic patterns. Shown in the third and fourth rows are the synthetically generated probability maps computed using the same re-sampling parameters - note the similarity to the estimated maps.

set: $s^\star = \arg\max_s M(p, s)$. If the similarity measure, $M(p, s^\star)$, is above a specified threshold, then a periodic pattern is assumed to be present in the estimated probability map, and the image is classified as re-sampled. This threshold is empirically determined using only the original images in the database to yield a false positive rate less than 1%.

With the ability to quantitatively measure whether an image has been re-sampled, we tested the efficacy of our technique to detecting a range of re-sampling parameters, and the sensitivity to simple counter-measures that may be used to hide traces of re-sampling. In these analyses we employed the same set of images as described in the beginning of this section, and used the same set of algorithmic parameters. The images were re-sampled using bi-cubic interpolation. The probability map for a re-sampled image was estimated and compared against a large set of synthetic maps. For up-sampling, 160 synthetic maps were generated with re-sampling rates between 1% and 100%, in steps of 0.6%. For down-sampling, 160 synthetic maps were generated with re-sampling rates between 1% and 50%, in steps of 0.3%. For rotations, 45 synthetic maps were generated with rotation angles between 1° and 45°, in steps of 1°.

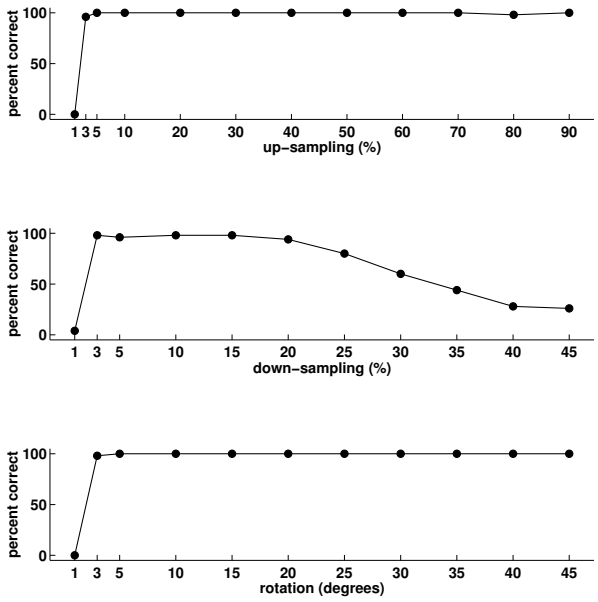


Fig. 12. Detection accuracy as a function of different re-sampling parameters. Each data point corresponds to the average detection accuracy from 50 images.

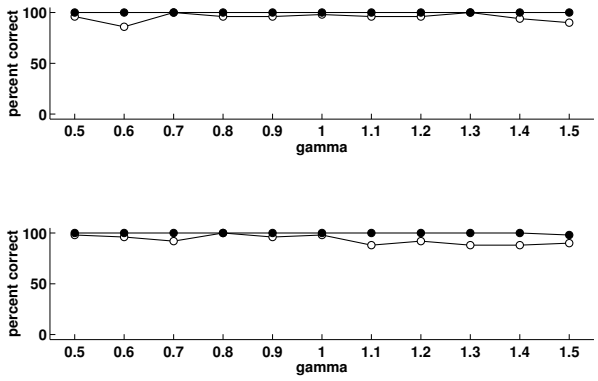


Fig. 13. Detection accuracy as a function of the amount of (non-linear) gamma correction. Shown in the top graph is the accuracy for up-sampling by a factor of 30% (black dots) and down-sampling by a factor of 20% (white dots). Shown below is the accuracy for rotating by 10° (black dots) and 2° (white dots). Each data point corresponds to the average detection accuracy from 50 images.

Shown in Fig. 12 are three graphs showing the detection accuracy for a range of up-sampling, down-sampling, and rotation rates. Each data point corresponds to the average detection accuracy from 50 images. In these results, the false-positive rate (an image incorrectly classified as re-sampled) is less than 1%. Note that detection is nearly perfect for up-sampling rates greater than 1%, and for rotations greater than 1°. As expected, the detection accuracy decreases as the down-sampling rate approaches 50%, Equation (15). We have also measured the detection accuracy in the presence of multiple re-samplings (e.g., up-sampling followed by rotation). In these cases, the detection accuracy is typically governed by the smallest detection accuracy of the multiple re-samplings.

Shown in Figs. 13-15 are graphs showing the robustness of our algorithm to simple counter-measures that may destroy the periodic correlations that result from re-sampling. Specifically, after re-sampling the image we (1) gamma corrected; (2) added noise to; or (3) JPEG compressed the image. Shown in each of these figures is the

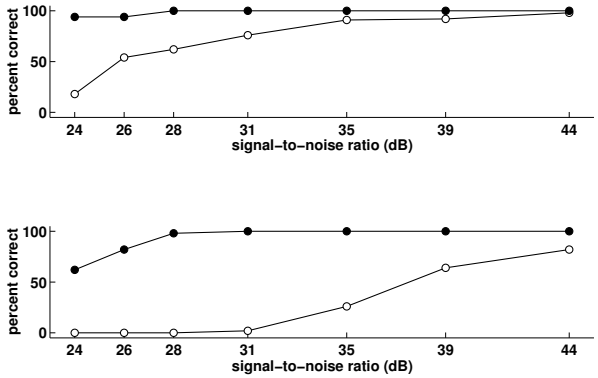


Fig. 14. Detection accuracy as a function of signal-to-noise ratio. Shown in the top graph is the accuracy for up-sampling by a factor of 30% (black dots) and down-sampling by a factor of 20% (white dots). Shown below is the accuracy for rotating by 10° (black dots) and 2° (white dots). Each data point corresponds to the average detection accuracy from 50 images.

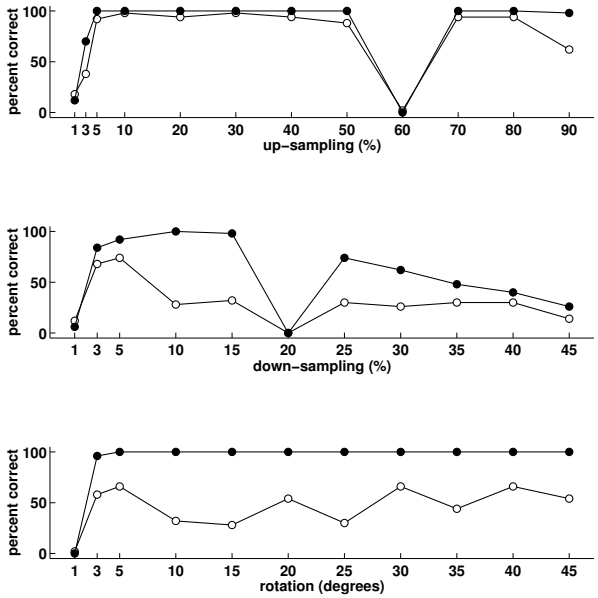


Fig. 15. Detection accuracy as a function of different re-sampling parameters and JPEG compression. The black dots correspond to a JPEG quality of 100 (out of 100), and the white dots to a quality of 97. Each data point corresponds to the average detection accuracy from 50 images.

detection accuracy for up-sampling by 30%, down-sampling by 20%, and rotating by 2° and 10°. Note that the detection is nearly perfect for a large range of gamma values, Fig. 13, and that detection accuracy remains reasonable for even fairly low signal-to-noise ratios, Fig. 14. Fig. 15, however, reveals a weakness in our approach. Shown here is the detection accuracy after the original TIFF image was JPEG compressed with a compression quality of 100 and 97 (out of 100). While the detection accuracy is good at a quality of 100, there is a precipitous fall in detection at a quality of 97 (at a quality of 90, detection is nearly at chance for all re-sampling rates). Note also that at an up-sampling rate of 60% and a down-sampling rate of 20% the detection accuracy drops suddenly. This is because the periodic JPEG blocking artifacts happen to coincide with the periodic patterns introduced by these re-sampling parameters - these artifacts do not interfere with the detection of rotations. The reason for the general

poor performance of detecting re-sampling in JPEG compressed images is two-fold. First, lossy JPEG compression introduces noise into the image (e.g., a compression quality of 90 introduces, on average, 28 db of noise), and as can be seen in Fig. 14, this amount of noise significantly affects the detection accuracy. Second, the block artifacts introduced by JPEG introduce very strong periodic patterns that interfere with and mask the periodic patterns introduced by re-sampling. In preliminary results, we found that under JPEG 2000 compression, detection remains robust down to 2 bits/pixel, with significant deterioration below 1.5 bits/pixel. This improved performance is most likely due to the lack of the blocking artifacts introduced by standard JPEG.

We have also tested our algorithm against GIF format images. Specifically, a 24-bit color (RGB) image was subjected to a range of re-samplings and then converted to 8-bit indexed color format (GIF). This conversion introduces approximately 21 db of noise. For rotations greater than 10°, up-sampling greater than 20%, and down-sampling greater than 15%, detection accuracy is, on average, 80%, 60%, and 30%, respectively, with a less than 1% false-positive rate. While not as good as the uncompressed TIFF images, these detection rates are roughly what would be expected with the level of noise introduced by GIF compression, Fig. 14. And finally, we have tested our algorithm against RGB images reconstructed from a color filter array (CFA) interpolation algorithm. In this case, the non-linear CFA interpolation does not interfere with our ability to detect re-sampling.

In summary, we have shown that for uncompressed TIFF images, and JPEG and GIF images with minimal compression we can detect whether an image region has been re-sampled (scaled, rotated, etc.), as might occur when an image has been tampered with.

IV. Discussion

When creating digital forgeries, it is often necessary to scale, rotate, or distort a portion of an image. This process involves re-sampling the original image onto a new lattice. Although this re-sampling process typically leaves behind no perceptual artifacts, it does introduce specific periodic correlations between the image pixels. We have shown how and when these patterns are introduced, and described a technique to automatically find such patterns in any region of an image. This technique is able to detect a broad range of re-sampling rates, and is reasonably robust to simple counter-attacks. This technique is not able to uniquely identify the specific re-sampling amount, as different re-samplings will manifest themselves with similar periodic patterns. Although we have only described how linear or cubic interpolation can be detected, there is no inherent reason why more sophisticated non-linear interpolation techniques (e.g., edge preserving interpolation) cannot be detected using the same basic framework of estimating local spatial correlations.

Our technique works in the complete absence of any digital watermark or signature, offering a complementary approach to authenticating digital images. While statistical techniques such as that presented here pose many challenges, we believe that their development will be important to contend with the cases when watermarking technologies are not applicable.

The major weakness of our approach is that it is currently only applicable to uncompressed TIFF images, and

JPEG and GIF images with minimal compression. We believe, however, that this technique will still prove useful in a number of different digital forensic settings - for example a court of law might insist that digital images be submitted into evidence in an uncompressed high-resolution format.

We are currently exploring several other techniques for detecting other forms of digital tampering. We believe that many complementary techniques such as that presented here, and those that we (e.g., [22]) and others develop, will be needed to reliably expose digital forgeries. There is little doubt that even with the development of a suite of detection techniques, more sophisticated tampering techniques will emerge, which in turn will lead to the development of more detection tools, and so on, thus making the creation of forgeries increasingly more difficult.

REFERENCES

[1] S. Katzenbeisser and F. Petitcolas, *Information Techniques for Steganography and Digital Watermarking*. Artec House, 2000.

[2] I. Cox, M. Miller, and J. Bloom, *Digital Watermarking*. Morgan Kaufmann Publishers, 2002.

[3] G. Friedman, "The trustworthy camera: Restoring credibility to the photographic image," *IEEE Transactions on Consumer Electronics*, vol. 39, no. 4, pp. 905–910, 1993.

[4] M. Schneider and S.-F. Chang, "A robust content-based digital signature for image authentication," in *IEEE International Conference on Image Processing*, vol. 2, 1996, pp. 227–230.

[5] D. Storck, "A new approach to integrity of digital images," in *IFIP Conference on Mobile Communication*, 1996, pp. 309–316.

[6] B. Macq and J.-J. Quisquater, "Cryptology for digital tv broadcasting," *Proceedings of the IEEE*, vol. 83, no. 6, pp. 944–957, 1995.

[7] S. Bhattacharjee and M. Kutter, "Compression-tolerant image authentication," in *IEEE International Conference on Image Processing*, vol. 1, 1998, pp. 435–439.

[8] C. Honsinger, P.Jones, M.Rabbani, and J. Stoffel, "Lossless recovery of an original image containing embedded data," U.S. Patent Application, Docket No. 77102/E-D, 1999.

[9] J. Fridrich, M. Goljan, and M. Du, "Invertible authentication," in *Proceedings of SPIE, Security and Watermarking of Multimedia Contents*, 2001.

[10] E. Lin, C. Podilchuk, and E. Delp, "Detection of image alterations using semi-fragile watermarks," in *Proceedings of SPIE, Security and Watermarking of Multimedia Contents II*, vol. 3971, 2000, pp. 52–163.

[11] 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, pp. 7/1–7/6.

[12] G.-J. Yu, C.-S. Lu, H.-Y. Liao, and J.-P. Sheu, "Mean quantization blind watermarking for image authentication," in *IEEE International Conference on Image Processing*, vol. 3, 2000, pp. 706–709.

[13] C.-Y. Lin and S.-F. Chang, "A robust image authentication algorithm surviving jpeg lossy compression," in *Proceedings of SPIE, Storage and Retrieval of Image/Video Databases*, vol. 3312, 1998, pp. 296–307.

[14] M. Yeung and F. Mintzer, "An invisible watermarking technique for image verification," in *Proceedings of the International Conference on Image Processing*, vol. 1, 1997, pp. 680–683.

[15] D. Kundur and D. Hatzinakos, "Digital watermarking for tell-tale tamper proofing and authentication," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1167–1180, 1999.

[16] M. U. Celik, G. Sharma, E. Saber, and A. M. Tekalp, "Hierarchical watermarking for secure image authentication with localization," *IEEE Transactions on Image Processing*, vol. 11, no. 6, pp. 585–595, 2002.

[17] J. Fridrich, "Security of fragile authentication watermarks with localization," in *Proceedings of SPIE, Electronic Imaging*, vol. 4675, 2002, pp. 691–700.

[18] J. Fridrich and M. Goljan, "Images with self-correcting capabilities," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 3, 1999, pp. 792–796.

[19] S. Craver, M. Wu, B. Liu, A. Stubblefield, B. Swartzlander, and D. Wallach, "Reading between the lines: Lessons from the SDMI challenge," in *10th USENIX Security Symposium*, Washington DC, 2001.

[20] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Prentice Hall, 1989.

[21] A. Dempster, N. Laird, and D. Rubin, "Maximum lilelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 99, no. 1, pp. 1–38, 1977.

[22] H. Farid and S. Lyu, "Higher-order wavelet statistics and their application to digital forensics," in *IEEE Workshop on Statistical Analysis in Computer Vision*, Madison, Wisconsin, 2003.

Appendix A: EM Algorithm

/* Initialize */

choose a random $\vec{\alpha}_0$

choose N and σ_0

set p_0 to the reciprocal of the range of the signal $\vec{y}$

set Y as in Equation (14)

set h to be a binomial low-pass filter of size $(N_h \times N_h)$

$n = 0$

repeat

/* expectation step */

for each sample i

$R(i) = \left| y(i) - \sum_{k=-N}^{N} \alpha_n(k) y(i+k) \right|$ /* residual */

end

$R = R \star h$ /* spatially average the residual error */

for each sample i

$P(i) = \frac{1}{\sigma_n \sqrt{2\pi}} e^{-R(i)^2/2\sigma_n^2}$ /* conditional probability */

$w(i) = \frac{P(i)}{P(i)+p_0}$ /* posterior probability */

end

/* maximization step */

W = 0

for each sample i

$W(i,i) = w(i)$ /* weighting matrix */

end

$\sigma_{n+1} = \left(\frac{\sum_i w(i) R^2(i)}{\sum_i w(i)} \right)^{1/2}$ /* new variance estimate */

$\vec{\alpha}_{n+1} = (Y^T W Y)^{-1} Y^T W \vec{y}$ /* new estimate */

$n = n + 1$

until ($\|\vec{\alpha}_n - \vec{\alpha}_{n-1}\| < \epsilon$) /* stopping condition */

Alin C Popescu received the B.E. degree in Electrical Engineering in 1999 from the University Politehnica of Bucharest, and the M.S. degree in Computer Science in 1999 from Université de Marne-la-Vallée. He is currently a Ph.D. candidate in Computer Science at Dartmouth College.

Hany Farid received the B.S. degree in Computer Science and Applied Mathematics in 1988 from the Unversity of Rochester, and then received the Ph.D. degree in 1997 in Computer Science from the University of Pennsylvania. He joined the Dartmouth faculty in 1999, following a two year post-doctoral position in Brain and Cognitive Sciences at the Massachusetts Institute of Technology.