Imaging Sensor Noise as Digital X-Ray for Revealing Forgeries

Mo Chen, Jessica Fridrich, Jan Lukáš, and Miroslav Goljan

Dept. of Electrical and Computer Engineering, SUNY Binghamton,
Binghamton, NY 13902-6000, USA
{mchen0,fridrich,jan.lukas,mgoljan}@binghamton.edu

Abstract. In this paper, we describe a new forensic tool for revealing digitally altered images by detecting the presence of photo-response non-uniformity noise (PRNU) in small regions. This method assumes that either the camera that took the image is available to the analyst or at least some other non-tampered images taken by the camera are available. Forgery detection using the PRNU involves two steps – estimation of the PRNU from non-tampered images and its detection in individual image regions. From a simplified model of the sensor output, we design optimal PRNU estimators and detectors. Binary hypothesis testing is used to determine which regions are forged. The method is tested on forged images coming from a variety of digital cameras and with different JPEG quality factors. The approximate probability of falsely identifying a forged region in a non-forged image is estimated by running the algorithm on a large number of non-forged images.

1 Introduction

The practice of forging photographs is probably as old as the art of photography itself. Digital photography and powerful image editing software make it very easy today to create believable forgeries of digital pictures even for a non-specialist. Verifying the content of digital images or identifying forged regions can be very crucial when digital pictures or video are presented as evidence in the court of law, for example, in child pornography and movie piracy cases (http://www.mpaa.org/piracy.asp) and even in cases involving scientific fraud [1,2].

Recently, several different methods for detecting digital forgeries were proposed [3–11]. For each of these methods, there are circumstances when they will fail to detect a forgery. For example, the copy-move detection method [9,10] is limited to one particular case of forgeries, when a certain part of the image was copied and pasted somewhere else in the same image (e.g., to cover an object). Methods based on detecting traces of resampling [6] or color filter array (CFA) interpolation artifacts [7] may produce less reliable results for processed images stored in the JPEG format. The method based on detection of inconsistencies in lighting [8] assumes nearly Lambertian surfaces for both the forged and original areas and might not work accurately when the object does not have a compatible surface, when pictures of both the origi-

nal and forged objects were taken under approximately similar lighting conditions, or during a cloudy day when no directional light source was present.

Detection of digital forgeries is a complex problem with no universally applicable solution. What is needed is a set of different tools that can be all applied to the image at hand. The decision about the content authenticity is then reached by interpreting the results obtained from different approaches. This accumulative evidence may provide a convincing enough argument that each individual method cannot.

In this paper, we describe another digital forensic tool by extending previous work on detection of forgeries [12] and employ the methodology recently proposed for camera identification [13]. The method localizes tampered image regions using the sensor pattern noise that each camera involuntarily inserts into each image as an authentication watermark. This approach is applicable whenever we are in a situation when the forged image is claimed to have been taken by a camera that we have in possession or, at least, we have other non-forged images taken by the camera. Because the pattern noise appears to be a unique stochastic fingerprint of digital imaging sensors [13], forged regions could be identified by verifying the consistency of their noise residual with the corresponding part of the pattern noise.

In the next section, we describe the sensor output model from which we derive in Section 3 and 4 an estimator and detector of the photo-response non-uniformity (PRNU). The pdf of the test statistics is obtained through a correlation predictor discussed in Section 5. The complete algorithm for forgery detection based on Neyman-Pearson hypothesis testing is detailed in Section 6. Experimental results are included in Section 7, while the last section contains a summary and discussion of limitations.

We use boldface font for vectors or matrices with $\mathbf{X}[i]$ denoting the i-th component of $\mathbf{X}$. Unless mentioned otherwise, all operations among vectors or matrices, such as product, ratio, or raising to a power, are *element-wise*. The norm of $\mathbf{X}$ is denoted as $\|\mathbf{X}\| = \sqrt{\mathbf{X} \odot \mathbf{X}}$ with $\mathbf{X} \odot \mathbf{Y} = \sum_{i=1}^{n} \mathbf{X}[i]\mathbf{Y}[i]$ being the dot product of two vectors. Denoting the sample means with a bar, the normalized correlation is

$$corr(\mathbf{X}, \mathbf{Y}) = \frac{(\mathbf{X} - \bar{\mathbf{X}}) \odot (\mathbf{Y} - \bar{\mathbf{Y}})}{\| \mathbf{X} - \bar{\mathbf{X}} \| \cdot \| \mathbf{Y} - \bar{\mathbf{Y}} \|}.$$

2 Sensor Output Model

Each digital camera contains a sensor that digitizes the image created by the optics by converting photons hitting each pixel to electrical signal. The signal then goes through a complex chain of processing that includes signal quantization, white balance, demosaicking, if the sensor is equipped with a CFA, color correction, gamma correction, filtering, and, optionally, JPEG compression. The processing details may vary greatly between cameras and are not always easily available.

In this section, we present a simplified model of in-camera processing [14] that includes the steps that are most relevant to our approach to forgery detection. We denote by $\mathbf{I}[i]$ the signal in one color channel at pixel i, $i = 1, \ldots, n$, generated by the sensor before demosaicking is applied and by $\mathbf{Y}[i]$ the incident light intensity at pixel

i. Here, we assume that the pixels are indexed, for example, in a row-wise manner and $n = n_1 n_2$, where $n_1 \times n_2$ are image dimensions. Dropping the pixel indices for better readability, we use the following model of the sensor output

$$\mathbf{I} = g^{\gamma} \cdot \left[(\mathbf{1} + \mathbf{K})\mathbf{Y} + \mathbf{\Theta}_n \right]^{\gamma} + \mathbf{\Theta}_q , \tag{1}$$

where g is the color channel gain, γ is the gamma correction factor (typically, $\gamma \approx 1/2.2$), $\mathbf{K}$ is a zero-mean multiplicative factor responsible for PRNU (the sensor fingerprint [13]), $\mathbf{\Theta}_q$ is the quantization noise and $\mathbf{\Theta}_n$ is a combination of various noise sources, such as dark current, shot noise, read-out noise, etc. [15,16]. The gain factor g adjusts the pixel intensity level according to the sensitivity of the pixel in the red, green, and blue spectral bands to obtain the correct white balance. We remind that all operations in (1) are element-wise.

We linearize (1) by factoring out the dominant term $\mathbf{Y}$ and leaving the first two terms in the Taylor expansion of $(1 + x)^{\gamma} = 1 + \gamma x + O(x^2)$

$$\mathbf{I} = \mathbf{I}^{(0)} + \gamma \mathbf{I}^{(0)} \mathbf{K} + \mathbf{\Theta} , \tag{2}$$

where we denoted $\mathbf{I}^{(0)} = (g\mathbf{Y})^{\gamma}$ the sensor output in the absence of noise; $\mathbf{\Theta}$ is a complex of independent random noise components.

3 PRNU Estimation

In this section, we describe the first step in our approach to forgery detection, which is the estimation of the PRNU $\mathbf{K}$ from a set of N images taken by the camera.

We first perform host signal rejection to improve the SNR between the signal of interest and observed data. The influence of the noiseless image $\mathbf{I}^{(0)}$ is suppressed by subtracting from both sides of (2) an estimate $\hat{\mathbf{I}}^{(0)} = F(\mathbf{I})$ of $\mathbf{I}^{(0)}$ obtained using a denoising filter[1] F

$$\mathbf{W} = \mathbf{I} - \hat{\mathbf{I}}^{(0)} = \gamma \mathbf{I}\mathbf{K} + \mathbf{I}^{(0)} - \hat{\mathbf{I}}^{(0)} + \gamma(\mathbf{I}^{(0)} - \mathbf{I})\mathbf{K} + \mathbf{\Theta} = \gamma \mathbf{I}\mathbf{K} + \mathbf{\Xi} . \tag{3}$$

The noise term $\mathbf{\Xi}$ contains $\mathbf{\Theta}$ and additional distortion introduced by the denoising filter.

Let $\mathbf{I}_1, \ldots, \mathbf{I}_N$ be N non-tampered images obtained by the camera. Assuming that the images are relatively smooth and non-saturated, the model (3) is approximately accurate. From (3), we have for each $k \in \{1, \ldots, N\}$

$$\frac{\mathbf{W}_k}{\gamma \mathbf{I}_k} = \mathbf{K} + \frac{\mathbf{\Xi}_k}{\gamma \mathbf{I}_k}, \quad \mathbf{W}_k = \mathbf{I}_k - \hat{\mathbf{I}}_k^{(0)}, \; \hat{\mathbf{I}}_k^{(0)} = F(\mathbf{I}_k) . \tag{4}$$

---

[1] We use a wavelet based denoising filter [17] that removes from images additive Gaussian noise with variance σ_F^2 (e.g., $\sigma_F^2 = 3$ for images with 256 levels of gray).

Under the assumption that for each pixel i the sequence $\Xi_1[i],\ldots,\Xi_N[i]$ is WGN (white Gaussian noise), the maximum likelihood estimate of $\mathbf{K}$ is (for detailed derivation and further discussion, see [13])

$$\hat{\mathbf{K}} = \frac{1}{\gamma}\frac{\sum_{k=1}^{N}\mathbf{W}_k\mathbf{I}_k}{\sum_{k=1}^{N}(\mathbf{I}_k)^2}, \tag{5}$$

which we calculate up to the multiplicative constant γ.

4 Detection of PRNU in Blocks

Having estimated the PRNU **K**, we identify tampered regions in an image by detecting the absence of PRNU in small blocks. Our basic assumption is that regions that have been tampered will not contain the PRNU from the camera. This is certainly true if the region has been copied from another image from a different camera. It is also true if the region is coming from an image obtained using the same camera as long as its spatial alignment in the image is different than in the forged image. We note that local "tampering" consisting of local image enhancement, such as contrast/brightness adjustment, sharpening, softening, or recoloring does not remove the PRNU and thus will not be detected by this forgery detection method.

If the forged image was modified using some known geometrical transformation, such as resizing or cropping, the PRNU must be pre-processed in the same manner before applying our forgery detection algorithm. If the geometrical operation is not known, the forgery detection algorithm might still apply after the geometrical transformation is identified. For this purpose, we might use the PRNU itself as a registration pattern or apply other forensic techniques, such as detection of resampling [6].

The forgery detection algorithm follows a similar structure as the method reported in [12]. In this paper, however, we use a more sophisticated algorithm that produces more reliable results. The presence of PRNU in block $\mathcal{B}$ is detected using binary hypothesis testing

$$\begin{aligned} &\mathrm{H}_0\text{: } \mathbf{W}[i] = \boldsymbol{\Xi}[i] \\ &\mathrm{H}_1\text{: } \mathbf{W}[i] = \tau\mathbf{I}[i]\hat{\mathbf{K}}[i] + \boldsymbol{\Xi}[i], \quad i \in \mathcal{B} \end{aligned} \tag{6}$$

where $\mathcal{B}$ is the index set characterizing the block. In (6), we assume that *within the tested block* $\boldsymbol{\Xi}$ is WGN with unknown mean and variance and τ is an unknown attenuation factor due to further processing that the image of interest might have been subjected to, such as kernel filtering, enhancement, or lossy compression.

The optimal detector for (6) is the normalized correlation (see, for example, [18]).

$$\rho = corr\left(\mathbf{I}\hat{\mathbf{K}}, \mathbf{W}\right). \tag{7}$$

where all signals in (7) are constrained to the block $\mathcal{B}$.

We can easily obtain the distribution of the test statistics ρ under hypothesis H_0 simply by correlating the known signal $\mathbf{I}[i]\hat{\mathbf{K}}[i]$, $i \in \mathcal{B}$, with noise residuals from other cameras. The distribution of ρ under H_1 is much harder to obtain. In spatially uniform and relatively smooth blocks, the statistical model in (6) is relatively accurate. However, in highly textured blocks, $\boldsymbol{\Xi}$ is not stationary or independent and the attenuation factor is not constant either because in such blocks the denoising filter is less successful in separating the image content and the noise (see (3)). To estimate the distribution of ρ under H_1 for a specific block, we construct a *predictor* of the test statistics as a function of selected factors that have a major influence on it. This predictor is obtained from blocks coming from a few non-tampered images from the same camera. In essence, the predictor tells us what the value of ρ and its distribution should be if the block was not tampered. We describe the predictor in the next section.

5 Correlation Predictor

In this section, we construct a predictor of the correlation ρ under H_1 on small blocks. From experiments, we determined that the most influential factors are image intensity, texture, and signal flattening.

The predictor is a mapping from some feature vector to a real number in the interval [0,1]—the predicted value of ρ. In order for the algorithm to have good localization properties, the block size should not be too large. However, it can not be too small either otherwise the correlation would have large variance. As a compromise, for typical sizes of digital camera images with 1 million pixels or more, square blocks with $b \times b$ pixels, $b = 128$, gave us quite good performance.

The correlation is higher in areas of high intensity because the PRNU signal $\mathbf{I}\hat{\mathbf{K}}$ is multiplicative. However, due to the finite dynamic range, it is not present in saturated regions ($\mathbf{I}[i] = 255$ for 8-bit per channel images) and is attenuated for $I_{crit} \le \mathbf{I}[i] \le 255$, where the critical value of intensity I_{crit} is typically in the range 240–250. Thus, we define the *intensity feature* $f_{\mathbf{I}}$ as the average attenuated image intensity

$$f_{\mathbf{I}} = \frac{1}{|\mathcal{B}|}\sum_{i \in \mathcal{B}} att(\mathbf{I}[i]), \tag{8}$$

where $att(x)$ is the attenuation function

$$att(\mathbf{I}[i]) = \begin{cases} e^{-(\mathbf{I}[i]-I_{crit})^2/\beta}, & \mathbf{I}[i] > I_{crit}, \\ \mathbf{I}[i]/I_{crit}, & \mathbf{I}[i] \le I_{crit}, \end{cases} \tag{9}$$

and β is a constant. For example, for our tested Canon G2 camera, we experimentally determined $I_{crit} = 250$, $\beta = 6$.

We calculate the *texture feature* $f_{\mathbf{T}}$ from the high-frequency component of the image. Since the denoising filter performs wavelet transform, we conveniently use this

intermediate data and generate a high-pass filtered image **F** as the inverse wavelet transform of the two outmost high-frequency wavelet subbands. The texture feature is then computed as

$$f_{\mathbf{T}} = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\frac{1}{1+\mathrm{var}_5(\mathbf{F}[i])}, \tag{10}$$

where $\mathrm{var}_5(\mathbf{F}[i])$ is the variance of **F** in the 5×5 neighborhood of the ith pixel. The reciprocal normalizes $f_{\mathbf{T}}$ to the interval [0, 1].

Image processing that is of low-pass filtering nature, such as JPEG compression, further attenuates the PRNU and thus decreases the correlation. In a relatively flat and high intensity unsaturated region, the predictor would thus incorrectly predict a high correlation. These "flattened" areas will typically have a low value of the local variance. Thus, we added the *flattening feature* $f_{\mathbf{S}}$ defined as the ratio of pixels in the block with average local variance above a certain threshold

$$f_{\mathbf{S}} = \frac{1}{|\mathcal{B}|}\left|\{i\in\mathcal{B} \mid \mathrm{var}_5(\mathbf{I}[i]) > c\mathbf{I}[i]+d\}\right|, \tag{11}$$

where c and d are appropriately chosen constants that depend on the sample variance of $\hat{\mathbf{K}}$ (e.g., c = 0.03, and d = 0.1 for Canon G2).

The correlation also strongly depends on the collective influence of texture and intensity. Sometimes, highly textured regions are also high-intensity regions. Thus, we included the following *texture-intensity feature*

$$f_{\mathbf{TI}} = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\frac{att(\mathbf{I}[i])}{1+\mathrm{var}_5(\mathbf{F}[i])}. \tag{12}$$

To capture the relationship between the features and the correlation, we chose a simple polynomial multivariate least square fitting because it is fast and gave us results comparable to more sophisticated tools, such as neural networks. We denote by $\boldsymbol{\rho}$ the column vector of K normalized correlations (7) calculated for K image blocks and $\mathbf{f}_{\mathbf{I}}$, $\mathbf{f}_{\mathbf{T}}$, $\mathbf{f}_{\mathbf{S}}$, and $\mathbf{f}_{\mathbf{TI}}$ the corresponding K-dimensional feature vectors. We model $\boldsymbol{\rho}$ as a linear combination of the 4 features and their 10 second-order terms

$$\boldsymbol{\rho}[k] = \theta_0 + \theta_1\mathbf{f}_{\mathbf{I}}[k] + \theta_2\mathbf{f}_{\mathbf{T}}[k] + \theta_3\mathbf{f}_{\mathbf{S}}[k] + \theta_4\mathbf{f}_{\mathbf{TI}}[k] + \theta_4\mathbf{f}_{\mathbf{I}}[k]\mathbf{f}_{\mathbf{I}}[k] + \theta_5\mathbf{f}_{\mathbf{I}}[k]\mathbf{f}_{\mathbf{T}}[k] + \ldots, \tag{13}$$

where $\boldsymbol{\theta} = (\theta_0, \theta_1, \ldots, \theta_{14})$ is the vector of 15 coefficients determined using the least square estimator $\boldsymbol{\theta} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\boldsymbol{\rho}$ and **H** is a K×15 matrix of features with a vector of ones in its first column. The estimated correlation is

$$\hat{\rho} = \left[1, f_{\mathbf{I}}, f_{\mathbf{T}}, f_{\mathbf{S}}, f_{\mathbf{TI}}, f_{\mathbf{I}}f_{\mathbf{I}}, f_{\mathbf{I}}f_{\mathbf{T}}, \ldots\right]\boldsymbol{\theta}. \tag{14}$$

In order to train the predictor, it is not necessary to use many images because one can extract a large number of overlapping blocks from a single image. In practice, good predictors can be obtained from as few as 8 images with diverse content. If the image under investigation is a JPEG image, it pays off to train the predictor on JPEG images

of approximately the same quality factor as it leads to more accurate forgery detection.

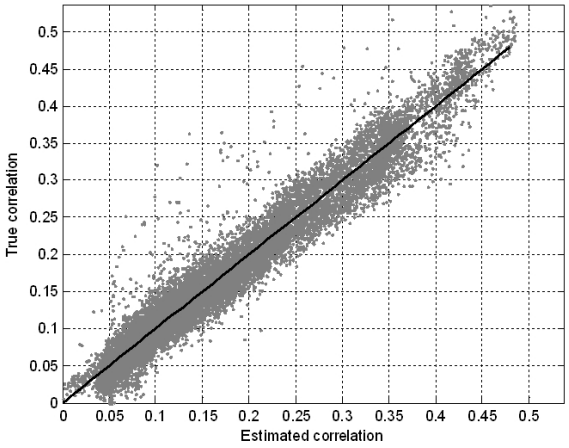


Fig. 1. Scatter plot ρ vs. $\hat{\rho}$ for K=30,000 128×128 blocks from 20 images from Canon G2.

6 Forgery Detection Algorithm

The algorithm starts by sliding a 128×128 block across the image and calculating the value of the test statistics $\rho_{\mathcal{B}}$ for each block $\mathcal{B}$. The pdf $p(x|\mathrm{H}_0)$ of $\rho_{\mathcal{B}}$ under H_0 is estimated by correlating the PRNU with noise residuals from other cameras and is modeled as generalized Gaussian (GG). For each block, the pdf $p(x|\mathrm{H}_1)$ is obtained from the predictor and is modeled again as GG. We basically fit the GG model with pdf $(\alpha / 2\sigma\Gamma(1/\alpha))e^{-(|x-\mu|/\sigma)^{\alpha}}$ through the data displayed in Fig. 1 with $\hat{\rho} \in (\rho_{\mathcal{B}} - \varepsilon, \rho_{\mathcal{B}} + \varepsilon)$ for some small $\varepsilon > 0$.

For each block $\mathcal{B}$, we first perform the Neyman-Pearson (NP) hypothesis testing by fixing the false alarm rate α. We decide that $\mathcal{B}$ has been tampered if $\rho_{\mathcal{B}} < Th$ and attribute this decision to the central pixel i of $\mathcal{B}$. The threshold Th is determined from the condition $\alpha = \int_{Th} p(x \mid \mathrm{H}_0)dx$. As a result, we obtain a $(n_1–127)\times(n_2–127)$ binary array $\mathbf{T}[i] = \rho_{\mathcal{B}}[i] < Th$ indicating the tampered pixels i with $\mathbf{T}[i] = 1$.

While calculating $\mathbf{T}$, we evaluate the p-values for each block (its central pixel i)

$$p[i] = \int_{-\infty}^{Th} p(x \mid \mathrm{H}_1)dx \tag{15}$$

which tell us how much we should trust our decision. We next remove from $\mathbf{T}$ tampered pixels i for which $p[i] > \beta$ and only label as tampered those pixels for which the p-value is smaller than β. The purpose of this step is to control falsely identified pixels as tampered. The resulting binary map identifying forged regions is what we call Digital X-ray. The forged objects show up as regions lacking the PRNU in the same

manner as bones show up in X-rays as they shield the radiation. The PRNU serves the same role as the X-rays.

The block dimensions impose a lower bound on the size of tampered regions that our algorithm can identify. Thus, we remove all simply connected regions from **T** that contain fewer than $b/2 \times b/2$ (64×64) pixels. Finally, we dilate the resulting binary map **T** with a square 20×20 kernel. The purpose of this final step is to compensate for the fact that we attribute the decision about the whole block only to its central pixel and thus potentially miss portions of the tampered boundary region.

7 Results

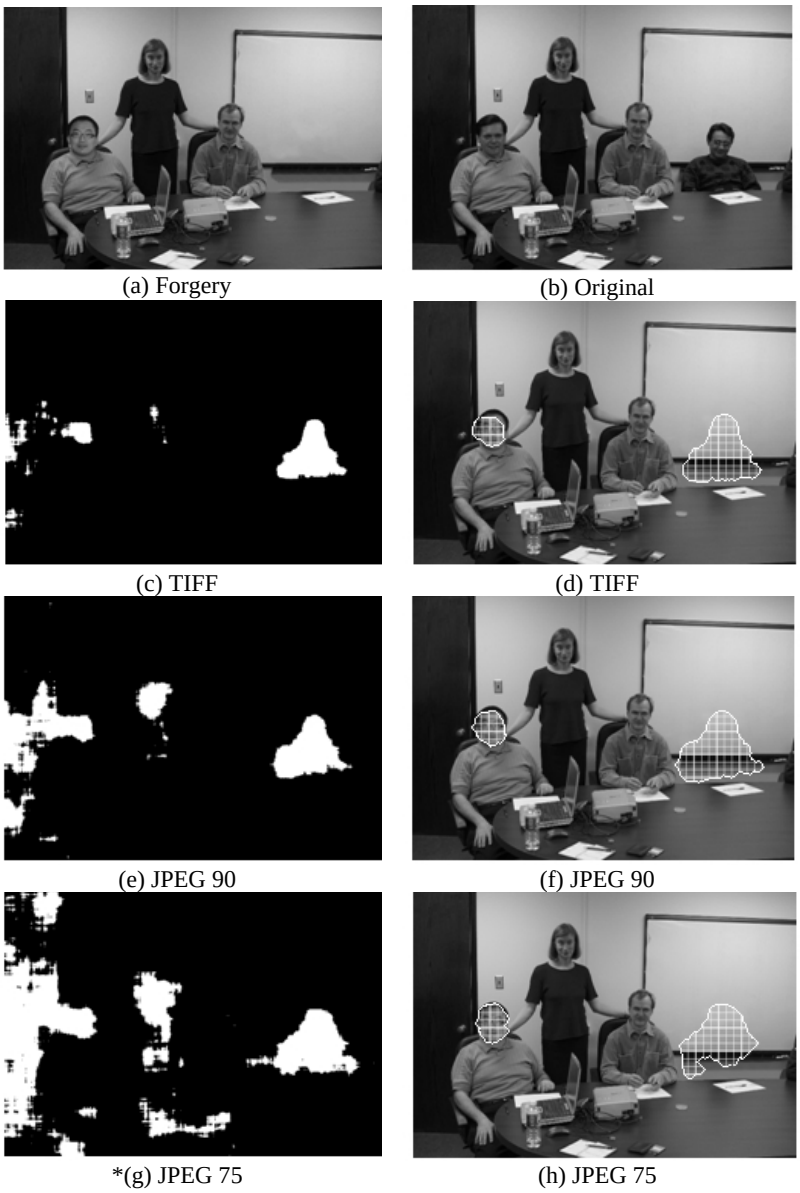
In this section, we subject our forgery detection algorithm to practical tests on forged images from 3 cameras: Canon G2 with a 4.1 megapixel (MP) CCD, Olympus C765 with a 4.1 MP CCD, and Olympus C3030 with a 3.3 MP CCD. We manipulated two images from each camera and stored them as TIFF and JPEG with quality factors 90 and 75. The forgeries varied from a simple copy-move within one image to object adding or removing. The PRNU was calculated from 30 blue sky images or uniformly lit test images obtained using a light box. If regular images were used, about 50 images would be required to have the PRNU of the same quality. The predictors were trained on more than 30,000 blocks from 20 regular images.

First, we calculated the test statistics for the unmatched cases by correlating 15,000 128×128 blocks from the PRNU with blocks from 100 images obtained using other cameras. A GG model was then fit through the data. The threshold Th was set to twice the standard deviation of the observed data. Since the GG fit was close to Gaussian for all tested cameras, this choice of threshold is equivalent to setting the false alarm rate α to approximately 3%. The threshold β for $p[i]$ was set to $\beta = 0.01$.

The performance of the proposed forgery detection technique is shown in Figs. 2–7, each of which includes the original image, the forged image, the NP decision result, and the forgery detection result (the X-ray) with tampered regions highlighted in the forged image.

As an implementation detail, we note that the sliding-window calculation of image features for the predictors can be computed efficiently using convolution implemented using FFT. For example, a full forgery X-ray of a 4 MP image takes between 4–5 minutes on a Pentium 3.4 GHz computer using Matlab.

To estimate the probability of false alarms for our method (detecting a tampered region in a non-forged image), we applied our algorithm to more than 400 non-forged images from the same three cameras in the JPEG format with quality 90. All false alarms have occurred in regions containing saturated background with dark regions, often combined with a complex texture, such as saturated sky shining through a mesh of black tree branches (two examples are shown in Fig. 8). Such regions naturally lack the PRNU and thus cannot be authenticated using our algorithm. Although not incorporated in this paper, these singularities could be removed by post-processing or expanding the predictor by adding a suitable feature.



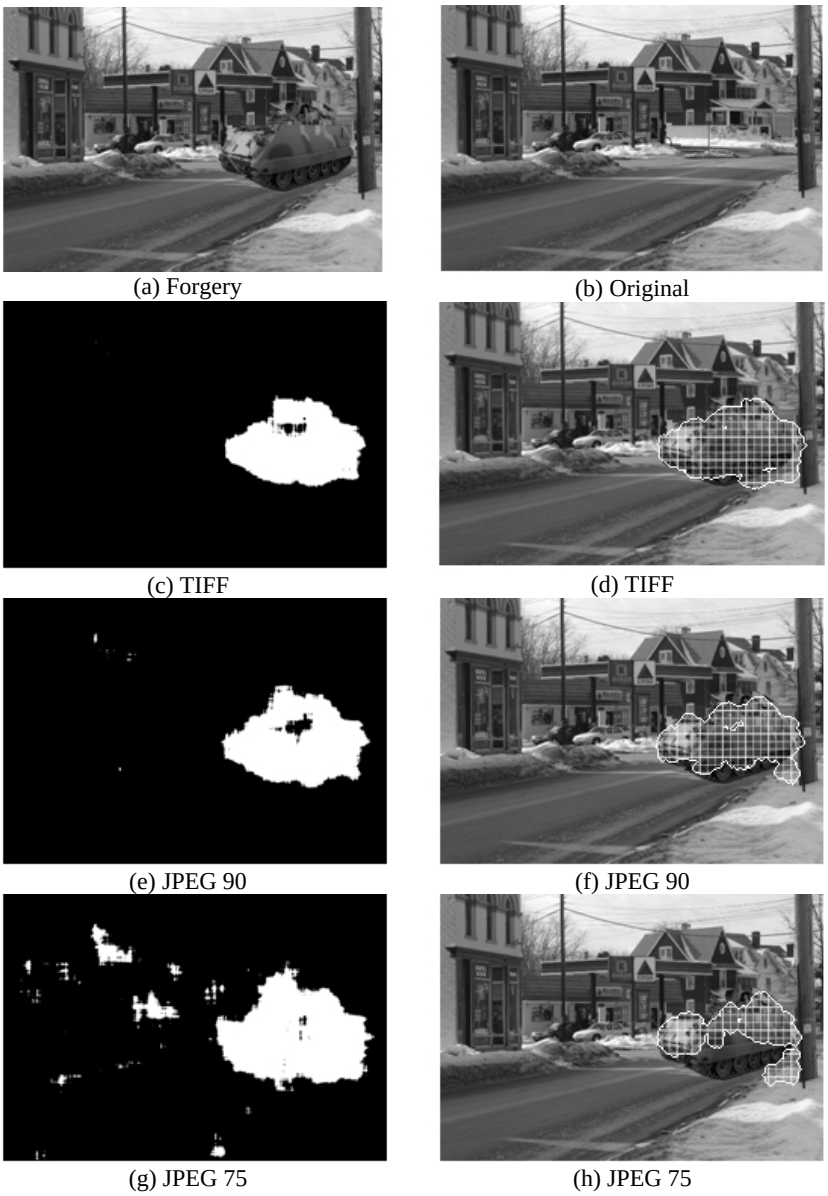
(a) Forgery (b) Original

(c) TIFF (d) TIFF

(e) JPEG 90 (f) JPEG 90

*(g) JPEG 75 (h) JPEG 75

Fig. 2. Forgery detection performance for a forged image from Canon G2 with $\alpha = 0.023$ and $\beta = 0.01$: (a) forged image; (b) original image; (c), (e), (g) is the NP decision for TIFF, JPEG 90, and JPEG 75, while (d), (f), (h) display the final forgery detection result.



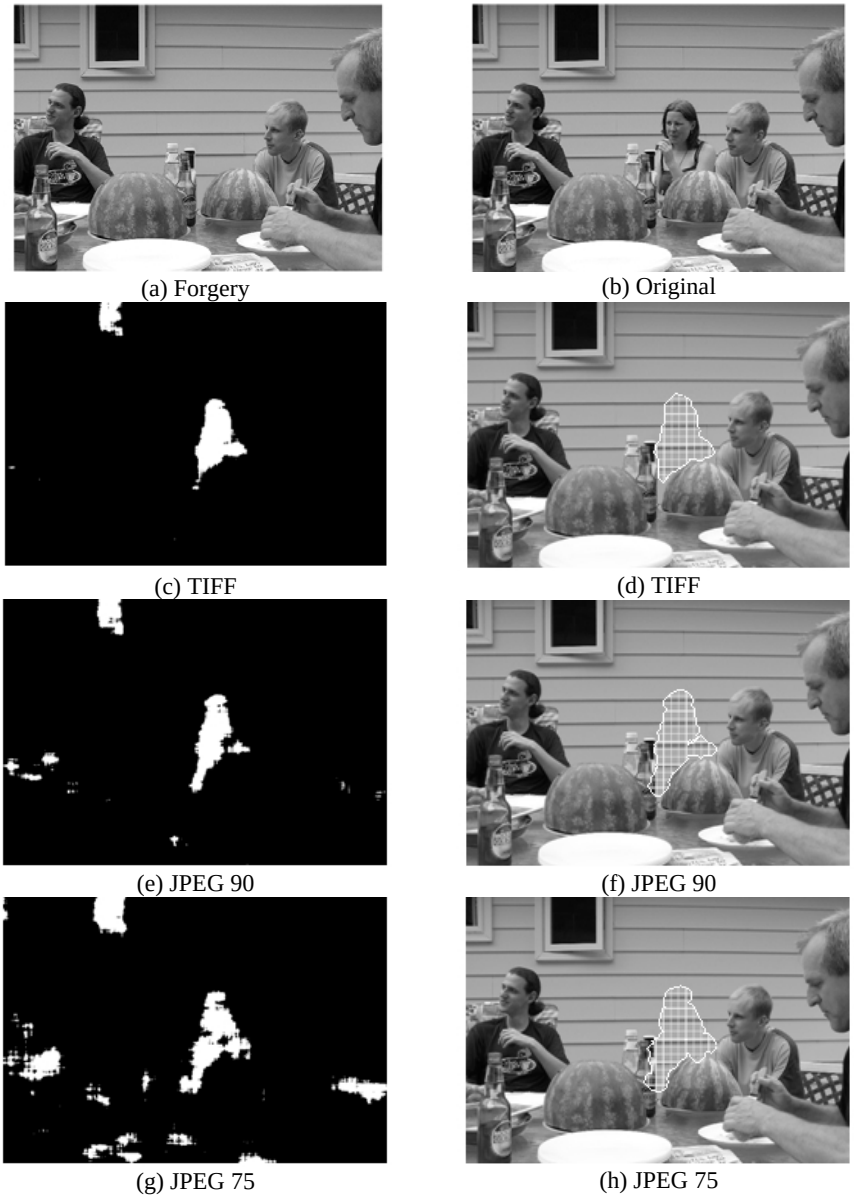
(a) Forgery (b) Original

(c) TIFF (d) TIFF

(e) JPEG 90 (f) JPEG 90

(g) JPEG 75 (h) JPEG 75

Fig. 3. Forgery detection performance for a forged image from Canon G2 with $\alpha = 0.023$ and $\beta = 0.01$. (a) forged image; (b) original image; (c), (e), (g) is the NP decision for TIFF, JPEG 90, and JPEG 75, while (d), (f), (h) display the final forgery detection result.



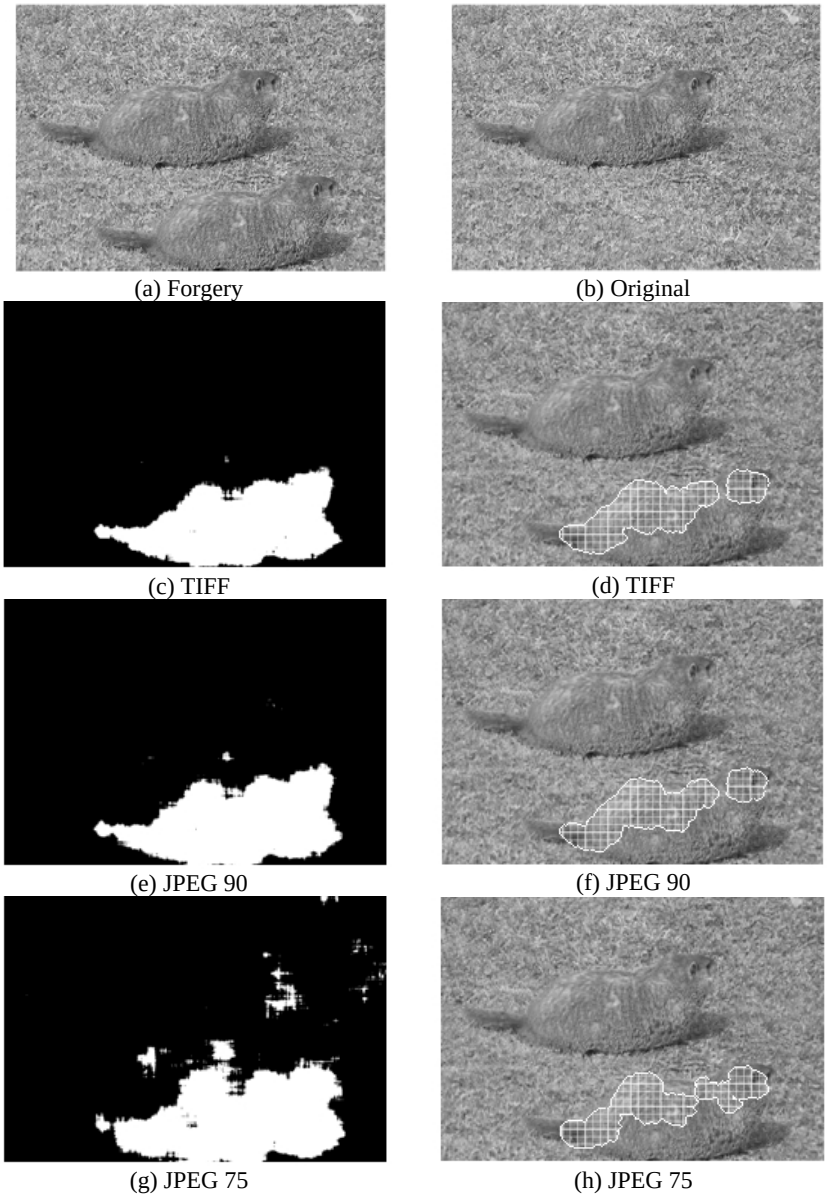
(a) Forgery (b) Original

(c) TIFF (d) TIFF

(e) JPEG 90 (f) JPEG 90

(g) JPEG 75 (h) JPEG 75

Fig. 4. Forgery detection performance for a forged image from Olympus C765 with $\alpha = 0.023$ and $\beta = 0.01$. (a) forged image; (b) original image; (c), (e), (g) is the NP decision for TIFF, JPEG 90, and JPEG 75, while (d), (f), (h) display the final forgery detection result.



(a) Forgery (b) Original

(c) TIFF (d) TIFF

(e) JPEG 90 (f) JPEG 90

(g) JPEG 75 (h) JPEG 75

Fig. 5. Forgery detection performance for a forged image from Olympus C765 with $\alpha = 0.023$ and $\beta = 0.01$. (a) forged image; (b) original image; (c), (e), (g) is the NP decision for TIFF, JPEG 90, and JPEG 75, while (d), (f), (h) display the final forgery detection result.

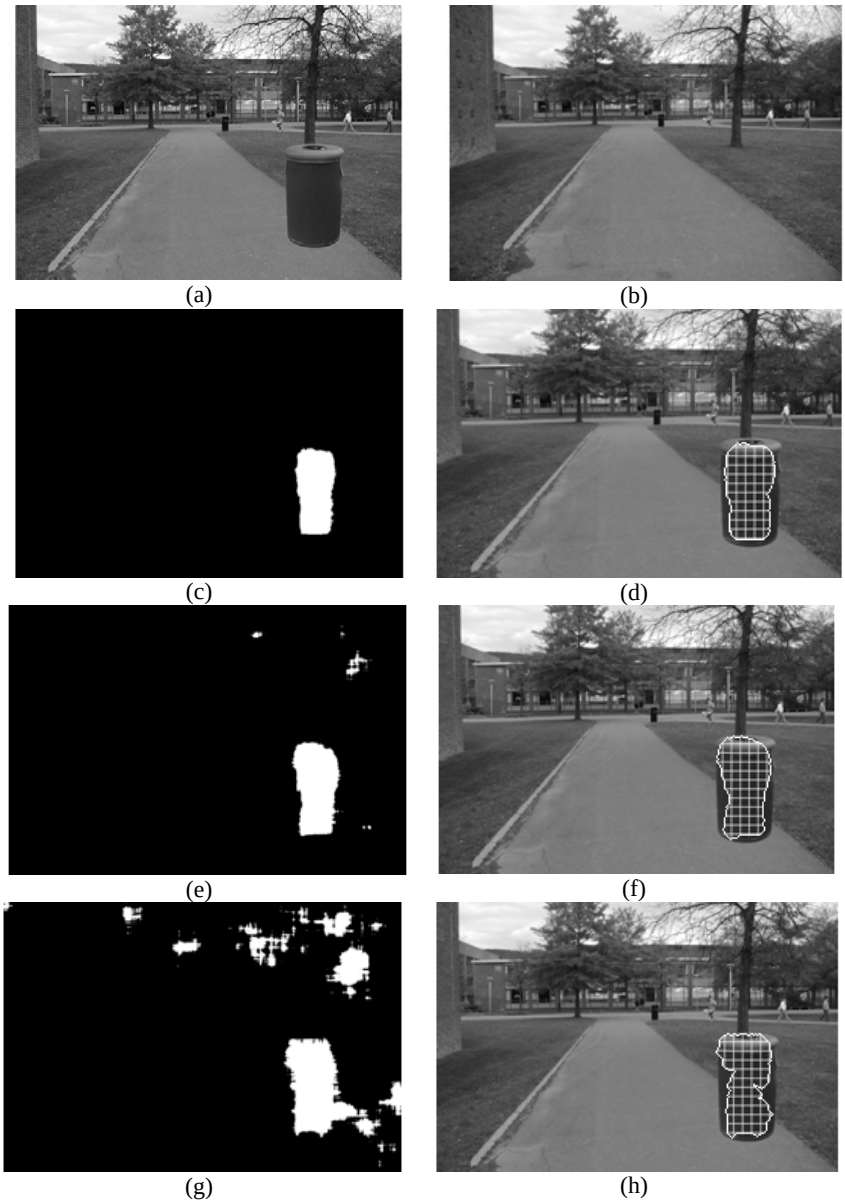
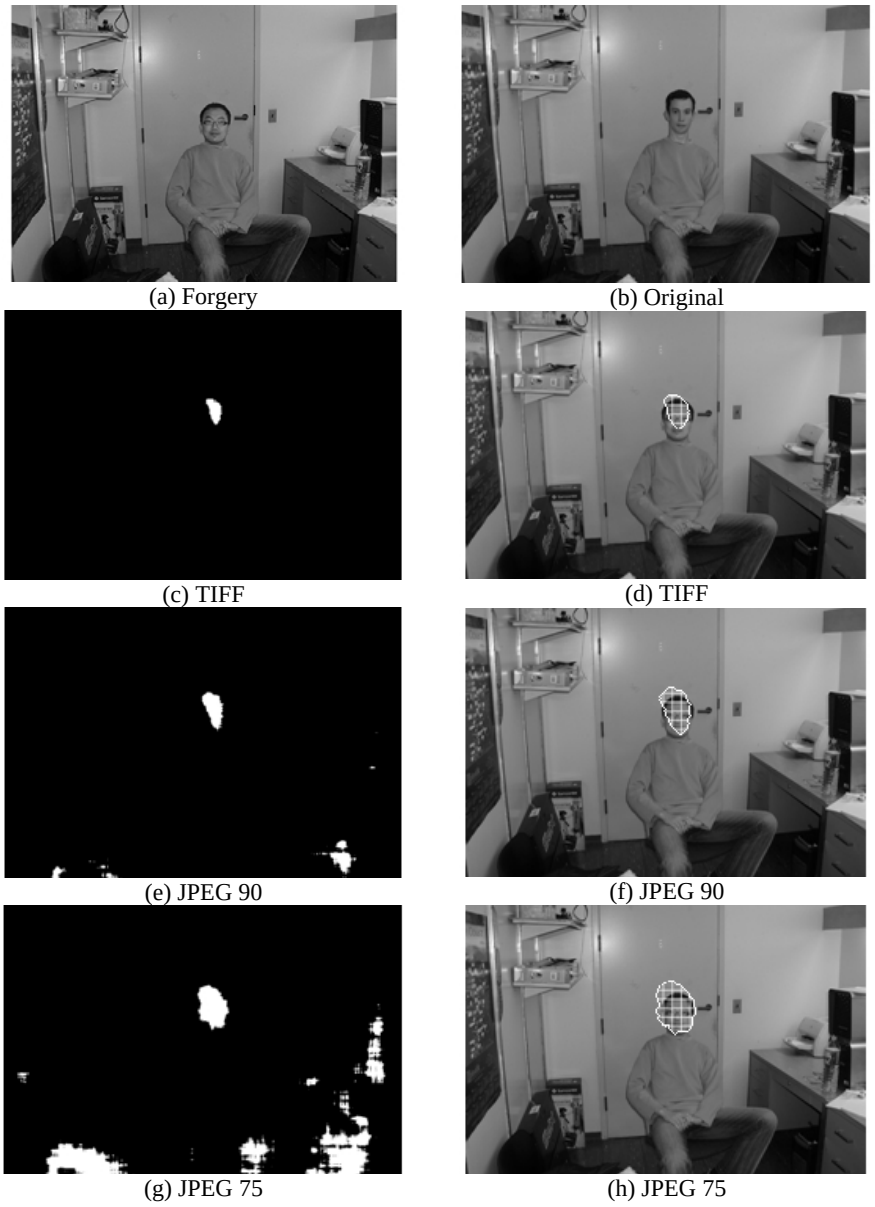


Fig. 6. Forgery detection performance for a forged image from Olympus C3030 with $\alpha = 0.023$ and $\beta = 0.01$. (a) forged image; (b) original image; (c), (e), (g) is the NP decision for TIFF, JPEG 90, and JPEG 75, while (d), (f), (h) display the final forgery detection result.



(a) Forgery (b) Original

(c) TIFF (d) TIFF

(e) JPEG 90 (f) JPEG 90

(g) JPEG 75 (h) JPEG 75

Fig. 7. Forgery detection performance for a forged image from Olympus C3030 with $\alpha = 0.023$ and $\beta = 0.01$. (a) forged image; (b) original image; (c), (e), (g) is the NP decision for TIFF, JPEG 90, and JPEG 75, while (d), (f), (h) display the final forgery detection result.

(a) (b)

Fig. 8. Representative examples of image patterns giving rise to false alarms. (a) and (b) are two non-forged images from Canon G2. The highlighted regions were falsely identified as tampered.

We next subjected the forgery detection algorithm to a large scale test in order to better evaluate its real performance. We prepared 345 forged images, all from Canon G2, into which we pasted rectangular regions from images taken using other cameras. The shape and the linear size of the pasted regions was selected randomly and no effort was made to make the forged regions look "naturally." The smallest and the largest sides of the rectangles were 228 and 512 pixels, respectively. All forgeries were saved with two JPEG quality factors – 90 and 75 and then inspected using the Digital X-ray algorithm while registering the ratio of correctly detected forged pixels and the ratio of falsely identified pixels (non-tampered pixels marked as tampered). Both ratios were calculated with respect to the size of the forged area.

Figure 9 shows the histograms of these two ratios expressed as percentage for all 345 tested images. For the JPEG quality factor 90, at least 2/3 of the forged region was correctly identified in 85% of forgeries. On the other hand, only 23% of forgeries contained more than 20% of falsely identified pixels (again with respect to the size of the forged region). For the JPEG quality factor 75, in 73% cases the X-ray correctly detected at least 2/3 of the forged area, while 21% of forgeries contained more than 20% of falsely identified pixels. The falsely identified areas were generally located around the boundary of the real forged area due to the 128×128 block size of the sliding window and the dilation post processing. We inspected all outliers and concluded that the few cases when a large portion of the tampered region was missed occurred when the pasted region contained a large dark region. The conservative values of thresholds in our algorithm are the reason why the region was not labeled as tampered because in such regions the PRNU is naturally suppressed. The few large false positives were all of the type already mentioned above and shown in Fig. 8.

8 Summary

In this paper, we described a new method for revealing digitally manipulated images. Assuming we have either the camera that took the image or some other non-tampered

images from the camera, we first estimate the photo-response non-uniformity, which serves as an authentication watermark. By detecting it in individual image blocks, one can localize the tampered region in the image. We use Neyman-Pearson hypothesis testing to identify the forged areas. The pdf of the test statistics is obtained from tests on images from other cameras (non-matched case) and using a correlation predictor (for the matched case). The proposed method can reliably identify forged areas larger than 64×64 pixels in JPEG images (tested with quality factor 75) while providing no falsely identified regions if regions that naturally lack the PRNU are excluded.

Among the potential future directions, we mention the possibility to construct a single predictor of the test statistics that would work for all cameras and calibrate it on a single non-forged image for a specific camera. This would further decrease the need for non-forged images taken with the same camera.

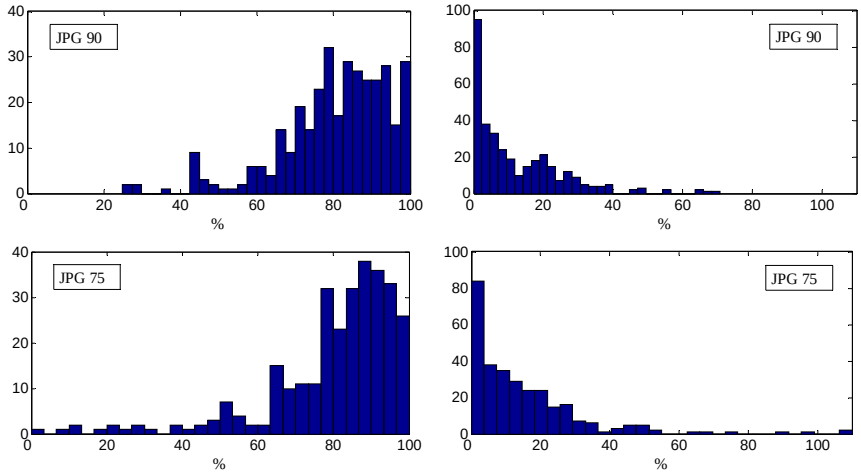


Fig. 9. Percentage of correctly identified tampered pixels (left) and falsely identified pixels (right) for 345 forged images from Canon G2 compressed using the JPEG with quality factor 90 (top) and 75 (bottom).

Acknowledgements

The work on this paper was supported by the AFOSR grant number FA9550-06-1-0046. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of Air Force Research Laboratory, or the U.S. Government.

References

1. Kennedy, D.: "Editorial retraction." Science, vol. **211**(5759) (2006) 335.
2. Pearson, H.: "Image manipulation: CSI: Cell biology." Nature, vol. **434** (2005) 952–953.
3. Ng, T.-T., Chang, S.-F., and Sun, Q.: "Blind Detection of Photomontage Using Higher Order Statistics." *IEEE International Symposium on Circuits and Systems,* vol. **5**, Vancouver, Canada (2004) v-688–v-691.
4. Avcibas, I., Bayram, S., Memon, N., Ramkumar, M., and Sankur, B.: "A Classifier Design for Detecting Image Manipulations." *IEEE Int. Conf. Image Proc.* vol. **4** (2004) 2645–2648.
5. Lin, Z., Wang, R., Tang, X., and Shum, H.-Y.: "Detecting Doctored Images Using Camera Response Normality and Consistency." *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. **1** (2005) 1087–1092.
6. Popescu, A.C. and Farid, H.: "Exposing Digital Forgeries by Detecting Traces of Resampling", *IEEE Transactions on Signal Processing*, vol. 53(2) (2005) 758–767.
7. Popescu, A.C. and Farid, H.: "Exposing Digital Forgeries in Color Filter Array Interpolated Images." *IEEE Transactions on Signal Processing*, vol. **53**(10) (2005) 3948–3959.
8. Johnson, M.K. and Farid, H.: "Exposing Digital Forgeries by Detecting Inconsistencies in Lighting." *Proc. ACM Multimedia and Security Workshop*. New York (2005) 1–9.
9. Fridrich, J., Soukal, D., and Lukáš, J.: "Detection of Copy-Move Forgery in Digital Images." *Proc. Digital Forensic Research Workshop*. Cleveland, August (2003).
10. Popescu, A.C. and Farid, H.: "Exposing Digital Forgeries by Detecting Duplicated Image Regions." *Technical Report,* TR2004-515. Dartmouth College, Computer Science (2004).
11. Farid, H.: "Exposing Digital Forgeries in Scientific Images." *Proc. ACM Multimedia & Security Workshop*. Geneva, Switzerland (2006) 29–36.
12. Lukáš, J., Fridrich, J., and Goljan, M.: "Detecting Digital Image Forgeries Using Sensor Pattern Noise." *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, vol. **6072**. San Jose, California (2006) 0Y1–0Y11.
13. Chen, M., Fridrich, J., and Goljan, M.: "Digital Imaging Sensor Identification (Further Study)." *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. **6505**. San Jose, California (2007) 0P–0Q.
14. Healey, G. and Kondepudy, R.: "Radiometric CCD Camera Calibration and Noise Estimation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. **16**(3) (1994) 267–276.
15. Janesick, J.R.: *Scientific Charge-Coupled Devices*. SPIE PRESS Monograph, vol. **PM83**, SPIE–The International Society for Optical Engineering (2001).
16. Holst, G.C.: *CCD Arrays, Cameras, and Displays*. 2nd edition. JCD Publishing & SPIE Pres, USA (1998).
17. Mihcak, M.K., Kozintsev, I., and Ramchandran, K.: "Spatially Adaptive Statistical Modeling of Wavelet Image Coefficients and its Application to Denoising." *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. **6**. Phoenix, Arizona (1999) 3253–3256.
18. Kay, S.M.: *Fundamentals of Statistical Signal Processing*. Volume II, Detection theory. Prentice Hall (1998).