Jessica Fridrich

Digital Image Forensics

Introducing methods that estimate and detect sensor fingerprint

This tutorial explains how photo-response nonuniformity (PRNU) of imaging sensors can be used for a variety of important digital forensic tasks, such as device identification, device linking, recovery of processing history, and detection of digital forgeries. The PRNU is an intrinsic property of all digital imaging sensors due to slight variations among individual pixels in their ability to convert photons to electrons. Consequently, every sensor casts a weak noise-like pattern onto every image it takes. This pattern, which plays the role of a sensor fingerprint, is essentially an unintentional stochastic spread-spectrum watermark that survives processing, such as lossy compression or filtering. This tutorial explains how this fingerprint can be estimated from images taken by the camera and later detected in a given image to establish image origin and integrity. Various forensic tasks are formulated as a two-channel hypothesis testing problem approached using the generalized likelihood ratio test. The performance of the introduced forensic methods is briefly illustrated on examples to give the reader a sense of the performance.

There exist two types of imaging sensors commonly found in digital cameras, camcorders, and scanners: charge-coupled device (CCD) and complementary metal-oxide semiconductor (CMOS). Both consist of a large number of photo detectors, also called pixels. Pixels are made of silicon and capture light by converting photons into electrons using the photoelectric effect. The accumulated charge is transferred out of the sensor, amplified, and then converted to a digital signal in an A/D converter and further processed before the data is stored in an image format, such as Joint Photographic Experts Group (JPEG).

Digital Object Identifier 10.1109/MSP.2008.931078

1053-5888/09/$25.00©2009IEEE

The pixels are usually rectangular and several microns across. The amount of electrons generated by the incident light at each pixel depends on the physical dimensions of the pixel photosensitive area and on the homogeneity of silicon. The pixels' physical dimensions vary slightly due to imperfections in the manufacturing process. Also, the inhomogeneity naturally present in silicon contributes to variations in quantum efficiency among pixels (the ability to convert photons to electrons). The differences among pixels can be captured with a matrix $\mathbf{K}$ of the same dimensions as the sensor. When the imaging sensor is illuminated with ideally uniform light intensity Y, in the absence of other noise sources, the sensor will register a noiselike signal $Y+Y\mathbf{K}$ instead. The term $Y\mathbf{K}$ is usually referred to as the PRNU.

BY DETECTING THE FINGERPRINT PRESENCE IN AN IMAGE, ONE CAN UNAMBIGUOUSLY DECIDE WHETHER THE IMAGE WAS PRODUCED BY A SPECIFIC CAMERA.

The matrix $\mathbf{K}$ is responsible for a major part of what we call the camera fingerprint. The fingerprint can be estimated experimentally, for example by taking many images of a uniformly illuminated surface and averaging the images to isolate the systematic component of all images. At the same time, the averaging suppresses random noise components, such as the shot noise (random variations in the number of photons reaching the pixel caused by quantum properties of light), the readout noise (random noise introduced during the sensor readout), and so on. The reader is referred to [1], [2] for a more detailed description of various noise sources affecting image acquisition. Figure 1 shows a magnified portion of a fingerprint from a four megapixel Canon G2 camera obtained by averaging 120 8-b gray-scale images with average gray-scale 128 across each image. Bright dots correspond to pixels that consistently generate more electrons, while dark dots mark pixels whose response is consistently lower. The variance in pixel values across the averaged image (before adjusting its range for visualization) was 0.5, or 51 dB. Although the strength of the fingerprint strongly depends on the camera model, the sensor fingerprint is typically quite a weak signal.

Figure 2 shows the magnitude of the Fourier transform of one pixel row in the averaged image. The signal resembles white noise with an attenuated high-frequency band.

Besides the PRNU, the camera fingerprint essentially contains all systematic defects of the sensor, including hot and dead pixels (pixels that consistently produce high and low output independently of illumination) and the so-called dark current (a noiselike pattern that the camera would take with its objective covered). But the most important component of the fingerprint is the PRNU. The PRNU term $Y\mathbf{K}$ is only weakly present in dark areas where $Y \approx 0$. Also, completely saturated areas of an image, where the pixels were filled to their full capacity, producing a constant signal, do not carry any traces of PRNU or any other noise for that matter.

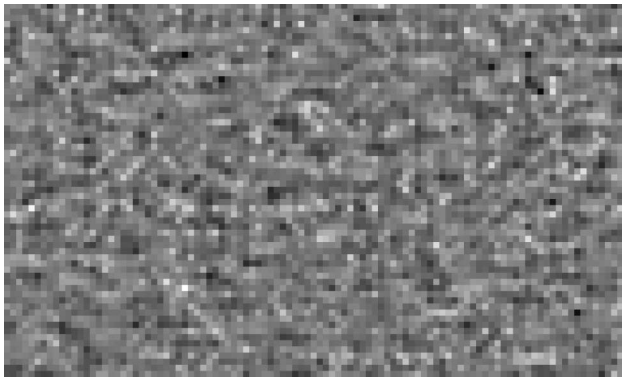
We note that essentially all imaging sensors (CCD, CMOS, JFET, or CMOS-Foveon-X3) are built from semiconductors, and their manufacturing techniques are similar. Therefore, these sensors will likely exhibit fingerprints with similar properties.

Even though the PRNU term is stochastic in nature, it is a relatively stable component of the sensor over its life span. The factor $\mathbf{K}$ is thus a very useful forensic quantity, responsible for a unique sensor fingerprint with the following important properties:

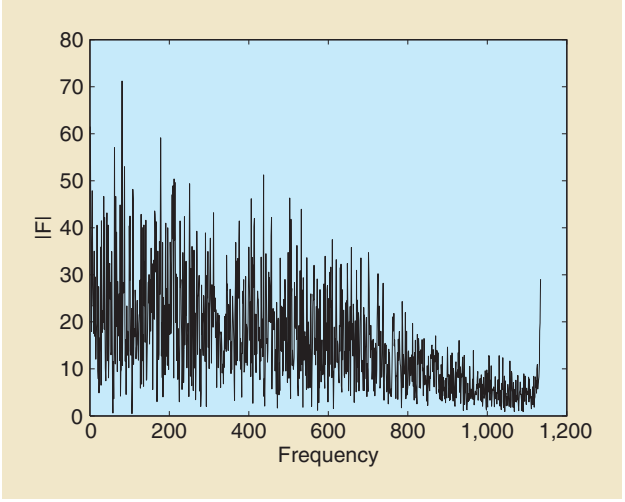
1) **Dimensionality:** The fingerprint is stochastic in nature and has a large information content, which makes it unique to each sensor.
2) **Universality:** All imaging sensors exhibit PRNU.
3) **Generality:** The fingerprint is present in every picture independently of the camera optics, camera settings, or scene content, with the exception of completely dark images.
4) **Stability:** It is stable in time and under a wide range of environmental conditions (temperature, humidity, etc.).
5) **Robustness:** It survives lossy compression, filtering, gamma correction, and many other typical processing procedures.

The fingerprint can be used for many forensic tasks:

- By testing the presence of a specific fingerprint in the image, one can achieve reliable device identification (e.g., prove that a certain camera took a given image) or



[FIG1] Magnified portion of the sensor fingerprint from Canon G2. The dynamic range was scaled to the interval [0, 255] for visualization.



[FIG2] Magnitude of Fourier transform of one row of the sensor fingerprint.

prove that two images were taken by the same device (device linking). The presence of camera fingerprint in an image is also indicative of the fact that the image under investigation is natural and not a computer rendering.

- By establishing the absence of the fingerprint in individual image regions, it is possible to discover replaced parts of the image. This task pertains to integrity verification.
- By detecting the strength or form of the fingerprint, it is possible to reconstruct some of the processing history. For example, one can use the fingerprint as a template to estimate geometrical processing, such as scaling, cropping, or rotation. Nongeometrical operations will also influence the strength of the fingerprint in the image and thus can potentially be detected.
- The spectral and spatial characteristics of the fingerprint can be used to identify the camera model or distinguish between a scan and a digital camera image (the scan will exhibit spatial anisotropy).

BESIDES DIGITAL CAMERAS, THE FORENSIC TECHNIQUE IS APPLICABLE TO CAMCORDERS AS WELL AS SCANNERS.

In this tutorial, we will explain the methods for estimating the fingerprint and its detection in images. The material is based on statistical signal estimation and detection theory.

The article is organized as follows. First, we describe a simplified sensor output model and use it to derive a maximum likelihood estimator for the fingerprint. At the same time, we point out the need to preprocess the estimated signal to remove certain systematic patterns that might increase false alarms in device identification and missed detections when using the fingerprint for image integrity verification. Taking up the sensor model again, the task of detecting the PRNU is formulated as a two-channel problem and approached using the generalized likelihood ratio test in the Neyman-Pearson setting. First, we derive the detector for device identification, and then we adapt it for device linking and fingerprint matching. We then show how the fingerprint can be used for integrity verification by detecting the fingerprint in individual image blocks. The reliability of camera identification and forgery detection using the sensor fingerprint is then illustrated with reference to real imagery.

In this article, boldface font will denote vectors (or matrices) of length specified in the text, e.g., $\mathbf{X}$ and $\mathbf{Y}$ are vectors of length n, and $\mathbf{X}[i]$ denotes the ith component of $\mathbf{X}$. Sometimes, we will index the pixels in an image using a two-dimensional index formed by the row and column index. Unless mentioned otherwise, all operations among vectors or matrices, such as product, ratio, raising to a power, and so on, are elementwise. The dot product of vectors is denoted as $\mathbf{X}\odot\mathbf{Y} = \sum_{i=1}^{n}\mathbf{X}[i]\mathbf{Y}[i]$ with $||\mathbf{X}|| = \sqrt{\mathbf{X}\odot\mathbf{X}}$ being the L_2 norm of $\mathbf{X}$. Denoting the sample mean with a bar, the normalized correlation is

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

SENSOR FINGERPRINT ESTIMATION

The PRNU is injected into the image during acquisition before the signal is quantized or processed in any other manner. In order to derive an estimator of the fingerprint, we need to formulate a model of the sensor output.

SENSOR OUTPUT MODEL

Even though the process of acquiring a digital image is quite complex and varies greatly across different camera models, some basic elements are common to most cameras. The light cast by the camera optics is projected onto the pixel grid of the imaging sensor. The charge generated through interaction of photons with silicon is amplified and quantized. Then, the signal from each color channel is adjusted for gain (scaled) to achieve proper white balance. Because most sensors cannot register color, the pixels are typically equipped with a color filter that lets only light of one specific color (red, green, or blue) enter the pixel. The array of filters is called the color filter array (CFA). To obtain a color image, the signal is interpolated or demosaicked. Finally, the colors are further adjusted to display correctly on a computer monitor through color correction and gamma correction. Cameras may also employ filtering techniques, such as denoising or sharpening. At the very end of this processing chain, the image is stored in the JPEG or some other format, which may involve quantization.

Let us denote by $\mathbf{I}[i]$ the quantized signal registered at pixel i, $i = 1, \ldots, m \times n$, before demosaicking. Here, $m \times n$ are image dimensions. Let $\mathbf{Y}[i]$ be the incident light intensity at pixel i. We drop the pixel indices for better readability and use the following vector form of the sensor output model:

$$\mathbf{I} = g^{\gamma}\cdot[(1+\mathbf{K})\mathbf{Y}+\Omega]^{\gamma}+\mathbf{Q}. \quad (1)$$

We remind the reader that all operations in (1) (and everywhere else in this tutorial) are elementwise. In (1), g is the gain factor (different for each color channel) and γ is the gamma correction factor (typically, $\gamma \approx 0.45$). The matrix $\mathbf{K}$ is a zero-mean noise-like signal responsible for the PRNU (the sensor fingerprint). Denoted by Ω is a combination of the other noise sources, such as the dark current, shot noise, and read-out noise [2]; $\mathbf{Q}$ is the combined distortion due to quantization and/or JPEG compression.

In parts of the image that are not dark, the dominant term in the square bracket in (1) is the scene light intensity, $\mathbf{Y}$. By factoring it out and keeping the first two terms in the Taylor expansion of $(1+x)^{\gamma} = 1+\gamma x + O(x^2)$ at $x = 0$, we obtain

$$\begin{aligned}\mathbf{I} &= (g\mathbf{Y})^{\gamma}\cdot[1+\mathbf{K}+\Omega/\mathbf{Y}]^{\gamma}+\mathbf{Q}\\ &\doteq (g\mathbf{Y})^{\gamma}\cdot(1+\gamma\mathbf{K}+\gamma\Omega/\mathbf{Y})+\mathbf{Q} = \mathbf{I}^{(0)}+\mathbf{I}^{(0)}\mathbf{K}+\Theta. \quad (2)\end{aligned}$$

In (2), we denoted $\mathbf{I}^{(0)} = (g\mathbf{Y})^{\gamma}$ the ideal sensor output in the absence of any noise or imperfections. Note that $\mathbf{I}^{(0)}\mathbf{K}$ is the PRNU term and $\Theta = \gamma\mathbf{I}^{(0)}\Omega/\mathbf{Y} + \mathbf{Q}$ is the modeling noise. In the last expression in (2), the scalar factor γ was absorbed into the PRNU factor $\mathbf{K}$ to simplify the notation.

SENSOR FINGERPRINT ESTIMATION

In this section, the above sensor output model is used to derive an estimator of the PRNU factor $\mathbf{K}$. A good introductory text on signal estimation and detection is [3] and [4].

The SNR between the signal of interest $\mathbf{I}^{(0)}\mathbf{K}$ and observed data $\mathbf{I}$ can be improved by suppressing the noiseless image $\mathbf{I}^{(0)}$ by subtracting from both sides of (2) a denoised version of $\mathbf{I}$, $\hat{\mathbf{I}}^{(0)} = F(\mathbf{I})$, obtained using a denoising filter F (see the appendix for a description of the filter used in the experiments in this tutorial)

$$\begin{aligned}\mathbf{W} &= \mathbf{I} - \hat{\mathbf{I}}^{(0)} = \mathbf{I}\mathbf{K} + \mathbf{I}^{(0)} - \hat{\mathbf{I}}^{(0)} + (\mathbf{I}^{(0)} - \mathbf{I})\mathbf{K} + \Theta \\ &= \mathbf{I}\mathbf{K} + \Xi\,. \end{aligned} \tag{3}$$

It is easier to estimate the PRNU term from $\mathbf{W}$ than from $\mathbf{I}$ because the filter suppresses the image content. We denoted by Ξ the sum of Θ and two additional terms introduced by the denoising filter.

Let us assume that we have a database of $d \geq 1$ images, $\mathbf{I}_1, \ldots, \mathbf{I}_d$, obtained by the camera. For each pixel i, the sequence $\Xi_1[i], \ldots, \Xi_d[i]$ is modeled as white Gaussian noise (WGN) with variance σ^2. The noise term is technically not independent of the PRNU signal $\mathbf{I}\mathbf{K}$ due to the term $(\mathbf{I}^{(0)} - \mathbf{I})\mathbf{K}$. However, because the energy of this term is small compared with $\mathbf{I}\mathbf{K}$, the assumption that Ξ is independent of $\mathbf{I}\mathbf{K}$ is reasonable.

From (3), we can write for each $k = 1, \ldots, d$

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

Under our assumption about the noise term, the log-likelihood of observing $\mathbf{W}_k/\mathbf{I}_k$ given $\mathbf{K}$ is

$$L(\mathbf{K}) = -\frac{d}{2}\sum_{k=1}^{d}\log(2\pi\sigma^2/(\mathbf{I}_k)^2) - \sum_{k=1}^{d}\frac{(\mathbf{W}_k/\mathbf{I}_k - \mathbf{K})^2}{2\sigma^2/(\mathbf{I}_k)^2}. \tag{5}$$

By taking partial derivatives of (5) with respect to individual elements of $\mathbf{K}$ and solving for $\mathbf{K}$, we obtain the maximum likelihood estimate $\hat{\mathbf{K}}$

$$\frac{\partial L(\mathbf{K})}{\partial \mathbf{K}} = \sum_{k=1}^{d}\frac{\mathbf{W}_k/\mathbf{I}_k - \mathbf{K}}{\sigma^2/(\mathbf{I}_k)^2} = 0 \Rightarrow \hat{\mathbf{K}} = \frac{\sum_{k=1}^{d}\mathbf{W}_k\mathbf{I}_k}{\sum_{k=1}^{d}(\mathbf{I}_k)^2}. \tag{6}$$

The Cramer-Rao lower bound (CRLB) gives us the bound on the variance of $\hat{\mathbf{K}}$

$$\frac{\partial^2 L(\mathbf{K})}{\partial \mathbf{K}^2} = -\frac{\sum_{k=1}^{d}(\mathbf{I}_k)^2}{\sigma^2} \Rightarrow \mathrm{var}(\hat{\mathbf{K}}) \geq \frac{1}{-E\left[\frac{\partial^2 L(\mathbf{K})}{\partial \mathbf{K}^2}\right]} = \frac{\sigma^2}{\sum_{k=1}^{d}(\mathbf{I}_k)^2}. \tag{7}$$

Because the sensor model (3) is linear, the CRLB tells us that the maximum likelihood estimator is minimum variance unbiased and its variance $\mathrm{var}(\hat{\mathbf{K}}) \sim 1/d$. From (7), we see that the best images for estimation of $\mathbf{K}$ are those with high luminance (but not saturated) and small σ^2 (which means smooth content). If the camera under investigation is in our possession, out-of-focus images of bright cloudy sky would be the best. In practice, good estimates of the fingerprint may be obtained from between 20 and 50 natural images, depending on the camera. If sky images are used instead of natural images, only one-half as many images (approximately) would be enough to obtain an estimate of the same accuracy.

THE ABSENCE OF THE FINGERPRINT IN A SMALL IMAGE REGION IS EVIDENCE OF TAMPERING.

The estimate $\hat{\mathbf{K}}$ contains all components that are systematically present in every image, including artifacts introduced by color interpolation, JPEG compression, on-sensor signal transfer [5], and sensor design. While the PRNU is unique to the sensor, the other artifacts are shared among cameras of the same model or sensor design. Consequently, PRNU factors estimated from two different cameras may be slightly correlated, which undesirably increases the false identification rate. Fortunately, the artifacts manifest themselves mainly as periodic signals in row and column averages of $\hat{\mathbf{K}}$ and can be suppressed simply by subtracting the averages from each row and column. For a PRNU estimate $\hat{\mathbf{K}}$ with m rows and n columns, the processing is described using the following pseudocode:

$r_i = 1/n \sum_{j=1}^{n}\hat{\mathbf{K}}[i,j]$

for $i = 1$ to m $\{\hat{\mathbf{K}}'[i,j] = \hat{\mathbf{K}}[i,j] - r_i \quad$ for $j = 1, \ldots, n\}$

$c_j = 1/m \sum_{i=1}^{m}\hat{\mathbf{K}}'[i,j]$

for $j = 1$ to n $\{\hat{\mathbf{K}}''[i,j] = \hat{\mathbf{K}}'[i,j] - c_j \quad$ for $i = 1, \ldots, m$.

The difference $\hat{\mathbf{K}} - \hat{\mathbf{K}}''$ is called the linear pattern (see Figure 3), and it is a useful forensic entity by itself—it can be used to classify a camera fingerprint to a camera model or brand. As this topic is not detailed in this tutorial, the reader is referred to [6] for more details.

To avoid cluttering the text with too many symbols, in the rest of this tutorial we will denote the processed fingerprint $\hat{\mathbf{K}}''$ with the same symbol $\hat{\mathbf{K}}$.

We close this section with a note on color images. In this case, the PRNU factor can be estimated for each color channel separately, thus obtaining three fingerprints of the same dimensions $\hat{\mathbf{K}}_R$, $\hat{\mathbf{K}}_G$, and $\hat{\mathbf{K}}_B$. Since these three fingerprints are highly correlated due to in-camera processing, in all forensic methods explained in this tutorial, before analyzing a color image under

investigation we convert it to gray scale and correspondingly combine the three fingerprints into one fingerprint using the usual conversion from RGB to gray scale:

$$\hat{\mathbf{K}} = 0.3\hat{\mathbf{K}}_R + 0.6\hat{\mathbf{K}}_G + 0.1\hat{\mathbf{K}}_B. \tag{8}$$

CAMERA IDENTIFICATION USING SENSOR FINGERPRINT

This section introduces general methodology for determining the origin of images or video using sensor fingerprint. We start with what we consider the most frequently occurring situation in practice, which is camera identification from images. Here, the task is to determine if an image under investigation was taken with a given camera. This is achieved by testing whether the image noise residual contains the camera fingerprint. Anticipating the next two closely related forensic tasks, we formulate the hypothesis testing problem for camera identification in a setting that is general enough to essentially cover the remaining tasks, which are device linking and fingerprint matching. In device linking, two images are tested to determine if they came from the same camera (the camera itself may not be available). The task of matching two estimated fingerprints occurs in matching two video clips because individual video frames from each clip can be used as a sequence of images from which an estimate of the camcorder fingerprint can be obtained (here, again, the cameras/camcorders may not be available to the analyst).

DEVICE IDENTIFICATION

We consider the scenario in which the image under investigation has possibly undergone a geometrical transformation, such as scaling or rotation. Let us assume that before applying any geometrical transformation the image was in gray scale represented with an $m \times n$ matrix $\mathbf{I}[i, j]$, $i = 1, \ldots, m, j = 1, \ldots, n$. Let us denote as $\mathbf{u}$ the (unknown) vector of parameters describing the geometrical transformation, $T_{\mathbf{u}}$. For example, $\mathbf{u}$ could be a scaling ratio or a two-dimensional vector consisting of the scaling parameter and unknown angle of rotation. In device identification, we wish to determine whether or not the transformed image

$$\mathbf{Z} = T_{\mathbf{u}}(\mathbf{I})$$

was taken with a camera with a known fingerprint estimate $\hat{\mathbf{K}}$. We will assume that the geometrical transformation is downgrading (such as downsampling) and thus it will be more advantageous to match the inverse transform $T_{\mathbf{u}}^{-1}(\mathbf{Z})$ with the fingerprint rather than matching $\mathbf{Z}$ with a downgraded version of $\hat{\mathbf{K}}$.

We now formulate the detection problem in a slightly more general form to cover all three forensic tasks mentioned above within one framework. The fingerprint detection is the following two-channel hypothesis testing problem:

$$\begin{aligned} \mathrm{H}_0&: \mathbf{K}_1 \neq \mathbf{K}_2 \\ \mathrm{H}_1&: \mathbf{K}_1 = \mathbf{K}_2 \end{aligned}, \tag{9}$$

where

$$\begin{aligned} \mathbf{W}_1 &= \mathbf{I}_1\mathbf{K}_1 + \Xi_1 \\ T_{\mathbf{u}}^{-1}(\mathbf{W}_2) &= T_{\mathbf{u}}^{-1}(\mathbf{Z})\mathbf{K}_2 + \Xi_2. \end{aligned} \tag{10}$$

In (10), all signals are observed with the exception of the noise terms Ξ_1 and Ξ_2 and the fingerprints $\mathbf{K}_1$ and $\mathbf{K}_2$. Specifically, for the device identification problem, $\mathbf{I}_1 \equiv 1$, $\mathbf{W}_1 = \hat{\mathbf{K}}$ estimated in the previous section, and Ξ_1 is the estimation error of the PRNU. $\mathbf{K}_2$ is the PRNU from the camera that took the image, $\mathbf{W}_2$ is the geometrically transformed noise residual, and Ξ_2 is a noise term. In general, $\mathbf{u}$ is an unknown parameter. Note that since $T_{\mathbf{u}}^{-1}(\mathbf{W}_2)$ and $\mathbf{W}_1$ may have different dimensions, the formulation (10) involves an unknown spatial shift between both signals, s.

Modeling the noise terms Ξ_1 and Ξ_2 as white Gaussian noise with known variances σ_1^2, σ_2^2, the generalized likelihood ratio test for this two-channel problem was derived in [7]. The test statistics is a sum of three terms, two energylike quantities and a cross-correlation term:

The complexity of evaluating these three expressions is proportional to the square of the number of pixels, $(m \times n)^2$, which makes this detector unusable in practice. Thus, we simplify this detector to a normalized cross-correlation (NCC) that can be evaluated using fast Fourier transform. Under H_1, the maximum in (11), which is shown at the bottom of the page, is mainly due to the contribution of the cross-correlation term, $C(\mathbf{u}, \mathbf{s})$, which exhibits a sharp peak for the proper values of the geometrical transformation. Thus, a much faster suboptimal detector is the NCC between $\mathbf{X}$ and $\mathbf{Y}$ maximized over all shifts s_1, s_2, and $\mathbf{u}$

$$\begin{aligned} t &= \max_{\mathbf{u},\mathbf{s}}\{E_1(\mathbf{u}, \mathbf{s}) + E_2(\mathbf{u}, \mathbf{s}) + C(\mathbf{u}, \mathbf{s})\}, \\ E_1(\mathbf{u}, \mathbf{s}) &= \sum_{i,j} \frac{\mathbf{I}_1^2[i,j](\mathbf{W}_1[i+s_1, j+s_2])^2}{\sigma_1^2\mathbf{I}_1^2[i,j] + \sigma_1^4\sigma_2^{-2}(T_{\mathbf{u}}^{-1}(\mathbf{Z})[i+s_1, j+s_2])^2} \\ E_2(\mathbf{u}, \mathbf{s}) &= \sum_{i,j} \frac{(T_{\mathbf{u}}^{-1}(\mathbf{Z})[i+s_1, j+s_2])^2(T_{\mathbf{u}}^{-1}(\mathbf{W}_2)[i+s_1, j+s_2])^2}{\sigma_2^2(T_{\mathbf{u}}^{-1}(\mathbf{Z})[i+s_1, j+s_2])^2 + \sigma_2^4\sigma_1^{-2}\mathbf{I}_1^2[i,j]} \\ C(\mathbf{u}, \mathbf{s}) &= \sum_{i,j} \frac{\mathbf{I}_1[i,j]\mathbf{W}_1[i,j](T_{\mathbf{u}}^{-1}(\mathbf{Z})[i+s_1, j+s_2])(T_{\mathbf{u}}^{-1}(\mathbf{W}_2)[i+s_1, j+s_2])}{\sigma_2^2\mathbf{I}_1^2[i,j] + \sigma_1^2(T_{\mathbf{u}}^{-1}(\mathbf{Z})[i+s_1, j+s_2])^2} \end{aligned} \tag{11}$$

$$\mathrm{NCC}[s_1, s_2; \mathbf{u}] = \frac{\sum_{k=1}^{m}\sum_{l=1}^{n}(\mathbf{X}[k,l]-\overline{\mathbf{X}})(\mathbf{Y}[k+s_1, l+s_2]-\overline{\mathbf{Y}})}{\|\mathbf{X}-\overline{\mathbf{X}}\|\|\mathbf{Y}-\overline{\mathbf{Y}}\|}, \tag{12}$$

which we view as an $m \times n$ matrix parameterized by $\mathbf{u}$, where

$$\mathbf{X} = \frac{\mathbf{I}_1\mathbf{W}_1}{\sqrt{\sigma_2^2\mathbf{I}_1^2 + \sigma_1^2(T_{\mathbf{u}}^{-1}(\mathbf{Z}))^2}}, \ \mathbf{Y} = \frac{T_{\mathbf{u}}^{-1}(\mathbf{Z})T_{\mathbf{u}}^{-1}(\mathbf{W}_2)}{\sqrt{\sigma_2^2\mathbf{I}_1^2 + \sigma_1^2(T_{\mathbf{u}}^{-1}(\mathbf{Z}))^2}}. \tag{13}$$

A more stable detection statistics, whose meaning will become apparent from error analysis later in this section and which we strongly advocate be used for all camera identification tasks, is the peak to correlation energy measure (PCE), defined as

$$\mathrm{PCE}(\mathbf{u}) = \frac{\mathrm{NCC}[\mathbf{s}_{\mathrm{peak}}; \mathbf{u}]^2}{\frac{1}{mn-|\mathcal{N}|}\sum_{\mathbf{s}, \mathbf{s}\notin\mathcal{N}}\mathrm{NCC}[\mathbf{s}; \mathbf{u}]^2}, \tag{14}$$

where for each fixed $\mathbf{u}$, $\mathcal{N}$ is a small region surrounding the peak value of NCC $\mathbf{s}_{\mathrm{peak}}$ across all shifts s_1, s_2.

For device identification from a single image, the fingerprint estimation noise Ξ_1 is much weaker compared to Ξ_2 for the noise residual of the image under investigation. Thus, $\sigma_1^2 = \mathrm{var}(\Xi_1) \ll \mathrm{var}(\Xi_2) = \sigma_2^2$ and (12) can be further simplified to a NCC between

$$\mathbf{X} = \mathbf{W}_1 = \hat{\mathbf{K}} \text{ and } \mathbf{Y} = T_{\mathbf{u}}^{-1}(\mathbf{Z})T_{\mathbf{u}}^{-1}(\mathbf{W}_2).$$

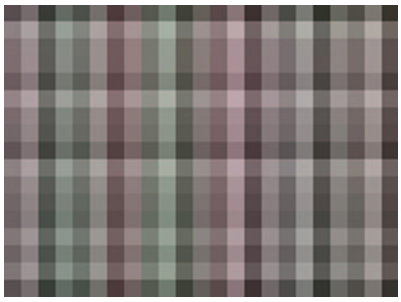
Recall that $\mathbf{I}_1 = 1$ for device identification when its fingerprint is known.

In practice, the maximum PCE value can be found by a search on a grid obtained by discretizing the range of $\mathbf{u}$. Unfortunately, because the statistics are noiselike for incorrect values of $\mathbf{u}$ and only exhibits a sharp peak in a small neighborhood of the correct value of $\mathbf{u}$, gradient methods do not apply and we are left with a potentially expensive grid search. The grid has to be sufficiently dense in order not to miss the peak. As an example, we now provide additional details about how one can carry out the search when $\mathbf{u} = r$ is an unknown scaling ratio. For more depth, the reader is advised to consult [9].

Assuming the image under investigation has dimensions $M \times N$, we search for the scaling parameter at discrete values $r_i \le 1$, $i = 0, 1, \ldots, R$, from $r_0 = 1$ (no scaling, just cropping) down to $r_{\min} = \max\{M/m, N/n\} < 1$

$$r_i = \frac{1}{1+0.005i}, \ i = 0, 1, 2, \ldots. \tag{15}$$

For a fixed scaling parameter r_i, the cross-correlation (12) does not have to be computed for all shifts $\mathbf{s}$ but only for those that move the upsampled image $T_{r_i}^{-1}(\mathbf{Z})$ *within* the dimensions of $\hat{\mathbf{K}}$ because only such shifts can be generated by cropping. Given that the dimensions of the upsampled image $T_{r_i}^{-1}(\mathbf{Z})$ are $M/r_i \times N/r_i$, we have the following range for the spatial shift $s = (s_1, s_2)$:



[FIG3] Detail of the linear pattern for Canon S40.

$$0 \le s_1 \le m - M/r_i \text{ and } 0 \le s_2 \le n - N/r_i. \tag{16}$$

The peak of the two-dimensional NCC across all spatial shifts $\mathbf{s}$ is evaluated for each r_i using PCE(r_i) (14). If $\max_i \mathrm{PCE}(r_i) > \tau$, we decide H_1 (camera and image are matched). Moreover, the value of the scaling parameter at which the PCE attains this maximum determines the scaling ratio r_{peak}. The location of the peak $\mathbf{s}_{\mathrm{peak}}$ in the normalized cross-correlation determines the cropping parameters. Thus, as a by-product of this algorithm, we can determine the processing history of the image under investigation (see Figure 4). The fingerprint can thus play the role of a synchronizing template similar to templates used in digital watermarking. It can also be used for reverse-engineering in-camera processing, such as digital zoom [9].

In any forensic application, it is important to keep the false alarm rate low. For camera identification tasks, this means that the probability P_{FA} that a camera that did not take the image is falsely identified must be below a certain user-defined threshold (Neyman-Pearson setting). Thus, we need to obtain a relationship between P_{FA} and the threshold on the PCE. Note that the threshold will depend on the size of the search space, which is in turn determined by the dimensions of the image under investigation.

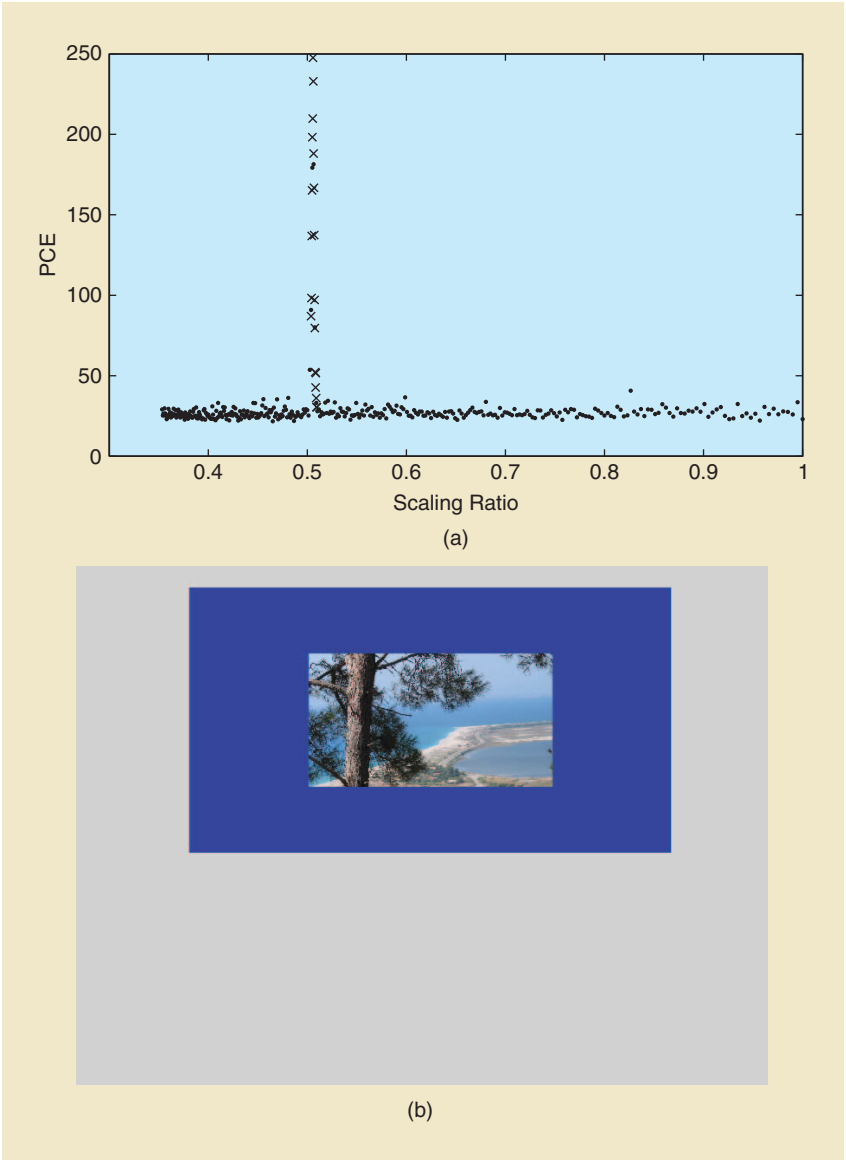
Under hypothesis H_0 for a fixed scaling ratio r_i, the values of the normalized cross-correlation $\mathrm{NCC}[\mathbf{s}; r_i]$ as a function of $\mathbf{s}$ are well modeled [9] as white Gaussian noise $\zeta^{(i)} \sim N(0, \sigma_i^2)$ with variance that may depend on i. Estimating the variance of the Gaussian model using the sample variance $\hat{\sigma}_i^2$ of $\mathrm{NCC}[\mathbf{s}; r_i]$ over $\mathbf{s}$ after excluding a small central region $\mathcal{N}$ surrounding the peak

$$\hat{\sigma}_i^2 = \frac{1}{mn-|\mathcal{N}|}\sum_{\mathbf{s}, \mathbf{s}\notin\mathcal{N}}\mathrm{NCC}[\mathbf{s}; r_i]^2, \tag{17}$$

we now calculate the probability p_i that $\zeta^{(i)}$ would attain the peak value $\mathrm{NCC}[\mathbf{s}_{\mathrm{peak}}; r_{\mathrm{peak}}]$ or larger by chance:

$$p_i = \int_{\mathrm{NCC}[\mathbf{s}_{\mathrm{peak}}; r_{\mathrm{peak}}]}^{\infty} \frac{1}{\sqrt{2\pi}\hat{\sigma}_i} e^{\frac{x^2}{2\hat{\sigma}_i^2}} dx = \int_{\hat{\sigma}_{\mathrm{peak}}\sqrt{\mathrm{PCE}_{\mathrm{peak}}}}^{\infty} \frac{1}{\sqrt{2\pi}\hat{\sigma}_i} e^{\frac{x}{2\hat{\sigma}_i^2}} dx$$
$$= Q\left(\frac{\hat{\sigma}_{\mathrm{peak}}}{\hat{\sigma}_i}\sqrt{\mathrm{PCE}_{\mathrm{peak}}}\right),$$

where $Q(x) = 1 - \Phi(x)$ with $\Phi(x)$ denoting the cumulative distribution function of a standard normal variable $N(0, 1)$ and $\mathrm{PCE}_{\mathrm{peak}} = \mathrm{PCE}(r_{\mathrm{peak}})$. As explained above, during the search for the



[FIG4] (a) Detected peak in PCE(r_i). (b) Visual representation of the detected cropping and scaling parameters $\mathbf{r}_{\text{peak}}$, $\mathbf{s}_{\text{peak}}$. The gray frame shows the original image size, while the blue frame shows the image size after cropping and before resizing.

cropping vector $\mathbf{s}$, we only need to search in the range (16), which means that we are taking maximum over $k_i = (m - M/r_i + 1) \times (n - N/r_i + 1)$ samples of $\zeta^{(i)}$. Thus, the probability that the maximum value of $\zeta^{(i)}$ would not exceed $\text{NCC}[\mathbf{s}_{\text{peak}}; r_{\text{peak}}]$ is $(1 - p_i)^{k_i}$. After R steps in the search, the probability of false alarm is

$$P_{\text{FA}} = 1 - \prod_{i=1}^{R} (1 - p_i)^{k_i}. \tag{18}$$

Since we can stop the search after the PCE reaches a certain threshold, we have $r_i \leq r_{\text{peak}}$. Because $\hat{\sigma}_i^2$ is nondecreasing in i, $\hat{\sigma}_{\text{peak}}/\hat{\sigma}_i \geq 1$. Because $Q(x)$ is decreasing, we have $p_i \leq Q(\sqrt{\text{PCE}_{\text{peak}}}) = p$. Thus, because $k_i \leq mn$, we obtain an upper bound on P_{FA}

$$P_{\text{FA}} \leq 1 - (1 - p)^{k_{\max}}, \tag{19}$$

where $k_{\max} = \sum_{i=0}^{R-1} k_i$ is the maximal number of values of the parameters r and $\mathbf{s}$ over which the maximum of (11) could be taken. Equation (19), together with $p = Q(\sqrt{\tau})$, determines the threshold for PCE, $\tau = \tau(P_{\text{FA}}, M, N, m, n)$.

This finishes the technical formulation and solution of the camera identification algorithm from a single image if the camera fingerprint is known. To provide the reader with some sense of how reliable this algorithm is, we include some experiments on real images later in this article. This algorithm can be used with small modifications for the other two forensic tasks formulated in the beginning of this section, which are device linking and fingerprint matching.

DEVICE LINKING

The detector derived in the previous section can be readily used with only a few changes for device linking or determining whether two images, $\mathbf{I}_1$ and $\mathbf{Z}$, were taken by the exact same camera [11]. Note that in this problem the camera or its fingerprint is not necessarily available.

The device-linking problem corresponds exactly to the two-channel formulation (9) and (10) with the GLRT detector (11). Its faster, suboptimal version is the PCE (14) obtained from the maximum value of $\text{NCC}[s_1, s_2; \mathbf{u}]$ over all $s_1, s_2; \mathbf{u}$ [see (12) and (13)]. In contrast to the camera identification problem, now the power of both noise terms, Ξ_1 and Ξ_2, is comparable and needs to be estimated from observations. Fortunately, because the PRNU term $\mathbf{IK}$ is much weaker than the modeling noise Ξ,reasonable estimates of the noise variances are simply $\hat{\sigma}_1^2 = \text{var}(\mathbf{W}_1)$, $\hat{\sigma}_2^2 = \text{var}(\mathbf{W}_2)$.

Also in contrast to the camera identification problem, the search for unknown scaling must now be enlarged to scalings $r_i > 1$ (upsampling) because the combined effect of unknown cropping and scaling for both images prevents us from easily identifying which image has been downscaled with respect to the other one. The error analysis carries over from the previous section.

Due to space limitations we do not include experimental verification of the device-linking algorithm. Instead, the reader is referred to [11].

MATCHING FINGERPRINTS

The last scenario, fingerprint matching, corresponds to the situation in which we need to decide whether or not two estimates of two potentially different fingerprints are identical. This

occurs, for example, in video-clip linking because the fingerprint can be estimated from all frames forming the clip [12].

The detector derived in the device identification section applies to this scenario as well. It can be further simplified because for matching fingerprints we have $\mathbf{I}_1 = \mathbf{Z} = 1$ and (12) simply becomes the normalized cross-correlation between $\mathbf{X} = \hat{\mathbf{K}}_1$ and $\mathbf{Y} = T_{\mathrm{u}}^{-1}(\hat{\mathbf{K}}_2)$.

For experimental verification of the fingerprint-matching algorithm for video clips, the reader is advised to consult [12].

THIS TUTORIAL INTRODUCES SEVERAL DIGITAL FORENSIC METHODS THAT CAPITALIZE ON THE FACT THAT EACH IMAGING SENSOR CASTS A UNIQUE NOISELIKE FINGERPRINT ON EVERY PICTURE IT TAKES.

FORGERY DETECTION USING CAMERA FINGERPRINT

A different, but nevertheless important, use of the sensor fingerprint is verification of image integrity. Certain types of tampering can be identified by detecting the fingerprint presence in smaller regions. The assumption is that if a region was copied from another part of the image (or an entirely different image), it will not have the correct fingerprint on it. The reader should realize that some content changes in the image may preserve the PRNU and will not be detected using this approach. A good example is changing the color of a stain to look like a bloodstain.

The forgery detection algorithm tests for the presence of the fingerprint in each $B \times B$ sliding block separately and then fuses all local decisions. For simplicity, we will assume that the image under investigation did not undergo any geometrical processing. For each block $\mathcal{B}_b$, the detection problem is formulated as hypothesis testing

$$\begin{aligned} \mathrm{H}_0&: \mathbf{W}_b = \Xi_b \\ \mathrm{H}_1&: \mathbf{W}_b = \mathbf{I}_b\hat{\mathbf{K}}_b + \Xi_b . \end{aligned} \tag{20}$$

Here, $\mathbf{W}_b$ is the block noise residual, $\hat{\mathbf{K}}_b$ is the corresponding block of the fingerprint, $\mathbf{I}_b$ is the block intensity, and Ξ_b is the modeling noise, assumed to be a white Gaussian noise with an unknown variance $\boldsymbol{\sigma}_{\Xi}^2$. The likelihood ratio test is the normalized correlation

$$\rho_b = \mathrm{corr}(\mathbf{I}_b\hat{\mathbf{K}}_b, \mathbf{W}_b). \tag{21}$$

In forgery detection, we may desire to control both types of error: failing to identify a tampered block as tampered and falsely marking a region as tampered. To this end, we will need to estimate the distribution of the test statistic under both hypotheses.

The probability density under H_0, $p(x|\mathrm{H}_0)$, can be estimated by correlating the known signal $\mathbf{I}_b\hat{\mathbf{K}}_b$ with noise residuals from other cameras. The distribution of ρ_b under H_1, $p(x|\mathrm{H}_1)$, is much harder to obtain because it is heavily influenced by the block content. Dark blocks will have lower values of correlation due to the multiplicative character of the PRNU. The fingerprint may also be absent from flat areas due to strong JPEG compression or saturation. Finally, textured areas will have lower values of correlation due to stronger modeling noise. This problem can be resolved by building a predictor of the correlation that will tell us what the value of the test statistic ρ_b and its distribution would be if the block b has not been tampered with and indeed comes from the camera.

The predictor is a mapping that needs to be constructed for each camera. The mapping assigns an estimate of the correlation ρ_b to each triple $(i_b,\ f_b,\ t_b)$, where the individual elements of the triple stand for a measure of intensity, saturation, and texture in block b. The mapping can be constructed, for example, using regression or machine-learning techniques by training them on a database of image blocks coming from images taken by the camera. The block size cannot be too small (because then the correlation ρ_b has too large a variance). On the other hand, large blocks would compromise the ability of the forgery detection algorithm to localize. Blocks of 64×64 or 128×128 pixels work well for most cameras.

A reasonable measure of intensity is the average intensity in the block

$$i_b = \frac{1}{|\mathcal{B}_b|}\sum_{i \in \mathcal{B}_b} \mathbf{I}[i]. \tag{22}$$

We take as a measure of flatness the relative number of pixels i, in the block whose sample intensity standard deviation $\sigma_{\mathbf{I}}[i]$ estimated from the local 3×3 neighborhood of i is below a certain threshold

$$f_b = \frac{1}{|\mathcal{B}_b|}\left|\{i \in \mathcal{B}_b\,|\sigma_{\mathbf{I}}[i] < c\mathbf{I}[i]\}\right|, \tag{23}$$

where $c \approx 0.03$ (for the Canon G2 camera). The best values of c vary with the camera model.

A good texture measure should somehow evaluate the amount of edges in the block. Among many available options, we give the following example:

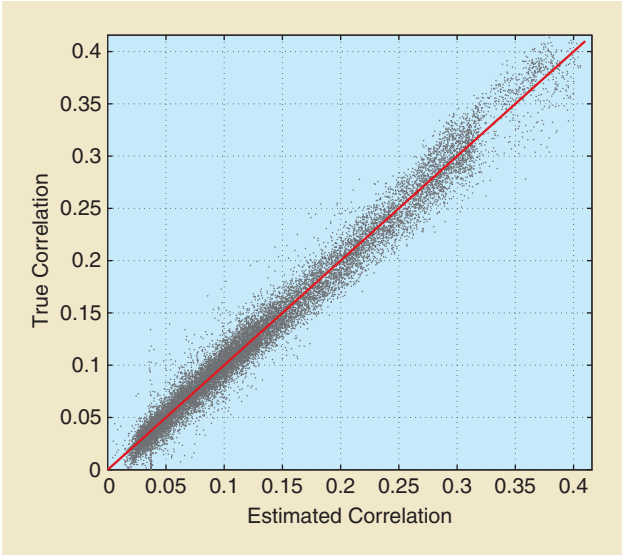
$$t_b = \frac{1}{|\mathcal{B}_b|}\sum_{i \in \mathcal{B}_b}\frac{1}{1 + \mathrm{var}_5(\mathbf{F}[i])}, \tag{24}$$

where $\mathrm{var}_5(\mathbf{F}[i])$ is the sample variance computed from a local 5×5 neighborhood of pixel i for a high-pass filtered version of the block, $\mathbf{F}[i]$, such as one obtained using an edge map or a noise residual in a transform domain.

Since one can obtain potentially hundreds of blocks from a single image, only a small number of images (e.g., ten) is needed to train (construct) the predictor. The data used for its construction can also be used to estimate the distribution of the prediction error ν_b

$$\rho_b = \hat{\rho}_b + \nu_b , \tag{25}$$

where $\hat{\rho}_b$ is the predicted value of the correlation.



[FIG5] Scatter plot of correlation ρ_b versus $\hat{\rho}_b$ for 30,000 128×128 blocks from 300 TIFF images produced by Canon G2.

Figure 5 shows the performance of the predictor constructed using second- order polynomial regression for a Canon G2 camera. Say that for a given block under investigation, we apply the predictor and obtain the estimated value $\hat{\rho}_b$. The distribution $p(x|\mathrm{H}_1)$ is obtained by fitting a parametric probability density function (pdf) to all points in Figure 7 whose estimated correlation is in a small neighborhood of $\hat{\rho}_b$, $(\hat{\rho}_b - \varepsilon, \hat{\rho}_b + \varepsilon)$. A sufficiently flexible model for the pdf that allows thin and thick tails is the generalized Gaussian model with pdf $\alpha/(2\sigma\Gamma(1/\alpha))e^{-(|x-\mu|/\sigma)^{\alpha}}$ with variance $\sigma^2\Gamma(3/\alpha)/\Gamma(1/\alpha)$, mean μ, and shape parameter α.

We now continue with the description of the forgery detection algorithm using sensor fingerprint. The algorithm proceeds by sliding a block across the image and evaluating the test statistic ρ_b for each block b. The decision threshold t for the test statistic ρ_b was set to obtain the probability of misidentifying a tampered block as nontampered $\Pr(\rho_b > t|\,\mathrm{H}_0) = 0.01$.

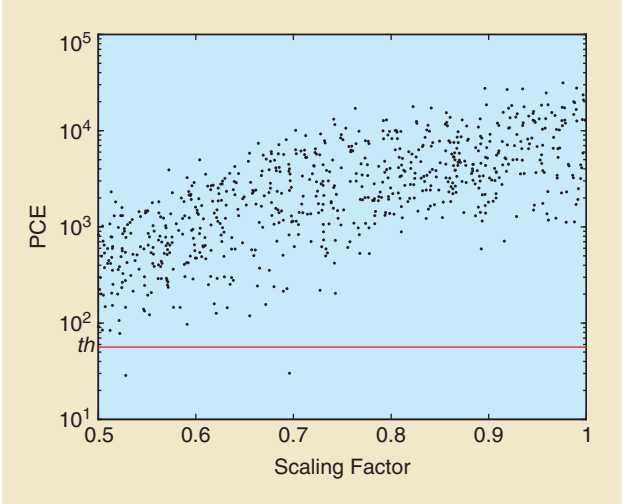
Block b is marked as potentially tampered if $\rho_b < t$, but this decision is attributed only to the central pixel i of the block. Through this process, for an $m \times n$ image we obtain an $(m–B+1) \times (n–B+1)$ binary array $\mathbf{Z}[i] = \rho_b < t$ indicating the potentially tampered pixels with $\mathbf{Z}[i] = 1$.

The above Neyman-Pearson criterion decides "tampered" whenever $\rho_b < t$ even though ρ_b may be "more compatible" with $p(x|\mathrm{H}_1)$, which is more likely to occur when ρ_b is small, such as for highly textured blocks. To control the amount of pixels falsely identified as tampered, we compute for each pixel i the probability of falsely labeling the pixel as tampered when it was not

$$p[i] = \int_{-\infty}^{t} p(x|\mathrm{H}_1)dx\,. \tag{26}$$

Pixel i is labeled as nontampered (we reset $\mathbf{Z}[i] = 0$) if $p[i] > \beta$, where β is a user-defined threshold (in experiments in this tutorial, $\beta = 0.01$). The resulting binary map $\mathbf{Z}$ identifies the forged regions in their raw form. The final map $\mathbf{Z}$ is obtained by further post-processing $\mathbf{Z}$.

The block size imposes a lower bound on the size of tampered regions that the algorithm can identify. We remove from $\mathbf{Z}$ all simply connected tampered regions that contain fewer than 64×64 pixels. The final map of forged regions is obtained by dilating $\mathbf{Z}$ with a square 20×20 kernel. The purpose of this step is to compensate for the fact that the decision about the whole block is attributed only to its central pixel, and we may thus miss portions of the tampered boundary region.



[FIG6] $\mathrm{PCE}_{\mathrm{peak}}$ as a function of the scaling ratio for 720 images matching the camera. The detection threshold τ, which is outlined with a horizontal line, corresponds to $P_{\mathrm{FA}} = 10^{-5}$.

EXPERIMENTAL VERIFICATION

In this section, we demonstrate how the forensic methods proposed in the previous two sections may be implemented in practice, and we include some sample experimental results to give the reader an idea of how the methods work with real imagery. The reader is referred to [9] and [13] for more extensive tests and to [11] and [12] for experimental verification of device linking and fingerprint matching for video clips. Camera identification from printed images is discussed in [10].

CAMERA IDENTIFICATION

A Canon G2 camera with a four-megapixel CCD sensor was used in all experiments in this section. The camera fingerprint was estimated for each color channel separately using the maximum likelihood estimator (6) from 30 blue-sky images acquired in the TIFF format. The estimated fingerprints were preprocessed using the column and row zero-meaning procedure explained above in the sensor fingerprint estimation section to remove any residual patterns not unique to the sensor. This step is very important because these artifacts would cause unwanted interference at certain spatial shifts and scaling factors and thus would decrease the PCE and substantially increase the false alarm rate.

The fingerprints estimated from all three color channels were combined into a single fingerprint using the linear

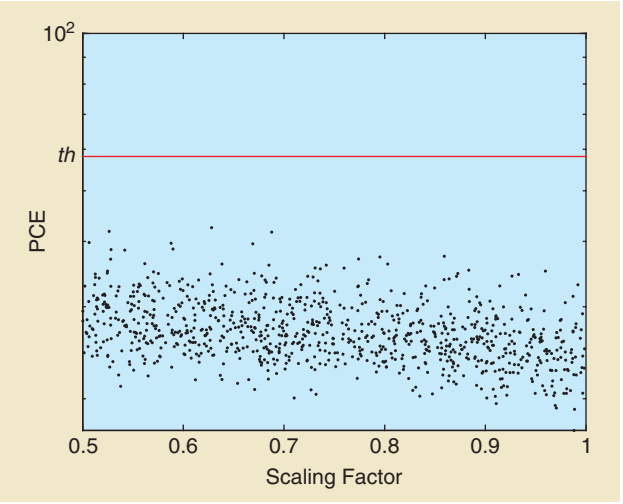
conversion rule used for conversion of color images to gray scale:

$$\hat{\mathbf{K}} = 0.3\hat{\mathbf{K}}_R + 0.6\hat{\mathbf{K}}_G + 0.1\hat{\mathbf{K}}_B.$$

All other images involved in this test were also converted to gray scale before applying the detectors described above in the device identification subsection of "Camera Identification Using Sensor Fingerprint."

The camera was further used to acquire 720 images, consisting of snapshots and various indoor and outdoor scenes under a wide spectrum of light conditions and zoom settings and spanning a period of four years. All images were taken at the full CCD resolution and with a high JPEG quality setting. Each image was first cropped by a random amount of up to 50% in each dimension. The upper-left corner of the cropped region was also chosen randomly, with uniform distribution within the upper-left quarter of the image. The cropped portion was subsequently downsampled by a randomly chosen scaling ratio $r \in [0.5, 1]$. Finally, the images were converted to gray scale and processed with 85% quality JPEG compression.

The detection threshold τ was chosen to obtain the probability of false alarm $P_{\text{FA}} = 10^{-5}$. The camera identification algorithm was run with $r_{\min} = 0.5$ on all images. Only two missed detections were encountered (Figure 6).



[FIG7] **PCE_{peak} for 915 images not matching the camera. The detection threshold τ is again outlined with a horizontal line and corresponds to $P_{\text{FA}} = 10^{-5}$.**

In the figure, the PCE is displayed as a function of the randomly chosen scaling ratio. The missed detections occurred for two highly textured images. In all successful detections, the cropping and scaling parameters were detected with accuracy better than two pixels in either dimension.

APPENDIX: DENOISING FILTER

The denoising filter used in the experimental sections of this tutorial is constructed in the wavelet domain. It was originally described in [22].

Let us assume that the image is a 512×512 gray scale image. Larger images can be processed in blocks, and color images are denoised for each color channel separately. The high-frequency wavelet coefficients of the noisy image are modeled as an additive mixture of a locally stationary i.i.d. (independent identically distributed) signal with zero mean (the noise-free image) and a stationary white Gaussian noise $N(0, \sigma_0^2)$ (the noise component). The denoising filter is built in two stages. In the first stage, we estimate the local image variance, while in the second stage the local Wiener filter is used to obtain an estimate of the denoised image in the wavelet domain. We now describe the individual steps:

1) Calculate the fourth-level wavelet decomposition of the noisy image with the eight-tap Daubechies quadrature mirror filters. We describe the procedure for one fixed level (it is executed for the high-frequency bands for all four levels). Denote the vertical, horizontal, and diagonal subbands as $h[i, j]$, $v[i, j]$, $d[i, j]$, where (i, j) runs through an index set $\mathcal{J}$ that depends on the decomposition level.

2) In each subband, estimate the local variance of the original noise-free image for each wavelet coefficient using the MAP estimation for four sizes of a square $W \times W$ neighborhood $\mathcal{N}$, for $W \in \{3, 5, 7, 9\}$.

$$\hat{\boldsymbol{\sigma}}_W^2[i, j] = \max\left(0, \frac{1}{W^2}\sum_{(i,j)\in\mathcal{N}} h^2[i, j] - \sigma_0^2\right), (i, j) \in \mathcal{J}.$$

Take the minimum of the four variances as the final estimate,

$$\hat{\boldsymbol{\sigma}}^2(i, j) = \min(\boldsymbol{\sigma}_3^2[i, j], \boldsymbol{\sigma}_5^2[i, j], \boldsymbol{\sigma}_7^2[i, j], \boldsymbol{\sigma}_9^2[i, j]), (i, j) \in \mathcal{J}.$$

3) The denoised wavelet coefficients are obtained using the Wiener filter

$$h_{\text{den}}[i, j] = h[i, j]\frac{\hat{\boldsymbol{\sigma}}^2[i, j]}{\hat{\boldsymbol{\sigma}}^2[i, j] + \sigma_0^2}$$

and similarly for $v[i, j]$, and $d[i, j]$, $(i, j) \in \mathcal{J}$.

4) Repeat steps 1–3 for each level and each color channel. The denoised image is obtained by applying the inverse wavelet transform to the denoised wavelet coefficients.

In all experiments, we used $\sigma_0 = 2$ (for dynamic range of images 0–255) to be conservative and to make sure that the filter extracts a substantial part of the PRNU noise even for cameras with a large noise component.

To test the false identification rate, we used 915 images, from more than 100 different cameras, downloaded from the Internet in native resolution. The images were cropped to four megapixels (the size of Canon G2 images) and subjected to the same random cropping, scaling, and JPEG compression as the 720 images discussed above. The threshold for the camera identification algorithm was set to the same value as in the previous experiment. All images were correctly classified as not coming from the tested camera (Figure 7). To experimentally verify the theoretical false alarm rate, millions of images would have to be examined, which is unfortunately not feasible.

THE FINGERPRINT CAN BE ESTIMATED FROM IMAGES TAKEN BY THE CAMERA BY AVERAGING THEIR NOISE COMPONENTS. THE DOMINANT SIGNAL IN THE FINGERPRINT IS THE PHOTO-RESPONSE NONUNIFORMITY (PRNU) CAUSED BY PIXELS' VARYING SENSITIVITY TO LIGHT.

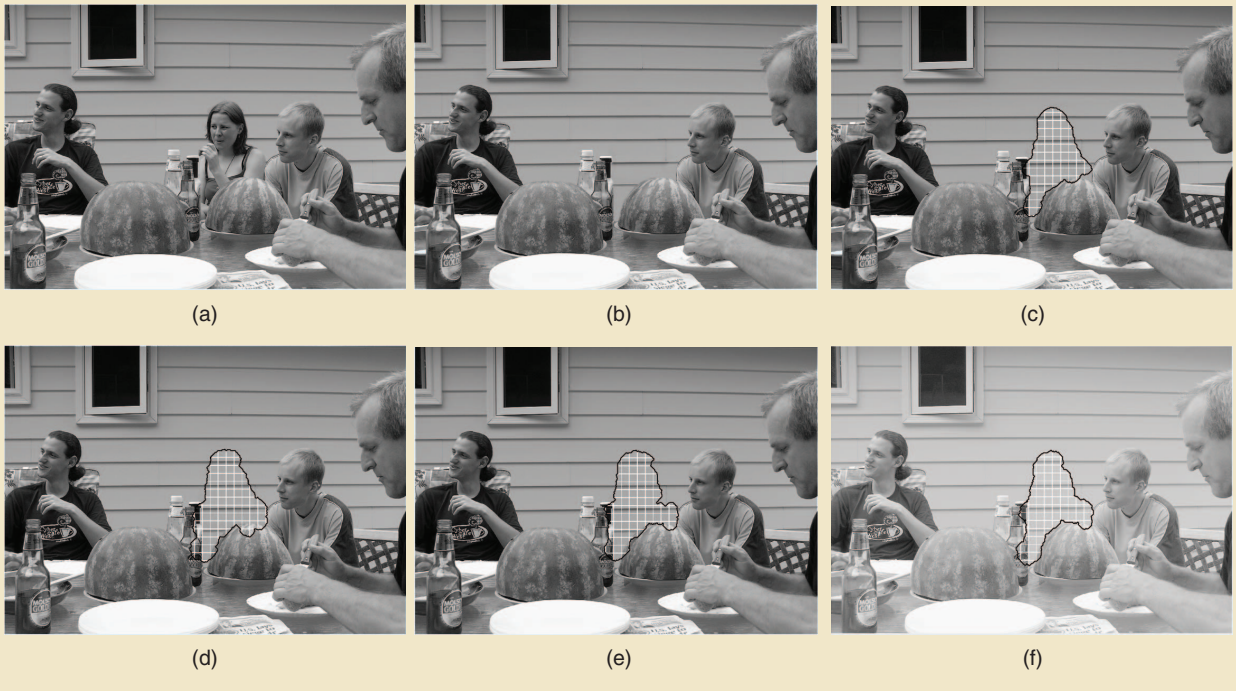
FORGERY DETECTION

Figure 8(a) shows the original image taken in the raw format by an Olympus C765 digital camera equipped with a four megapixel CCD sensor. Using Photoshop, the girl in the middle was covered by pieces of the house siding from the background [Figure 8(b)]. The forged image was then stored in the TIFF and JPEG (with 75% quality JPEG compression) formats. The corresponding output of the forgery detection algorithm, shown in Figures 8(c) and (d), is the binary map $\mathbf{Z}$ highlighted using a square grid. The last two figures show the map $\mathbf{Z}$ after the forgery was subjected to denoising using a 3×3 Wiener filter [Figure 8(e)] followed by 90% quality JPEG compression and when the forged image was processed using gamma correction with $\gamma = 0.5$ and again saved using JPEG 90% quality compression [Figure 8(f)]. In all cases, the forged region was accurately detected.

More examples of forgery detection using this algorithm, including the results of tests on a large number of automatically created forgeries as well as nonforged images, can be found in the original publication [13].

CONCLUSIONS

This tutorial introduces several digital forensic methods that capitalize on the fact that each imaging sensor casts a noiselike fingerprint on every picture it takes. The main component of the fingerprint is the PRNU, which is caused by pixels' varying capability to convert light to electrons. Because the differences among pixels are due to imperfections in the manufacturing process and silicon inhomogeneity, the fingerprint is essentially a stochastic, spread-spectrum signal and thus robust to distortion.



[FIG8] An (a) original and (b) forged Olympus C765 image and its detection from a forgery stored as (c) TIFF, (d) 75% quality JPEG, denoised using a 3×3 Wiener filter and saved as (e) 90% quality JPEG, and (f) gamma-corrected with $\gamma = 0.5$ and stored as 90% quality JPEG.

Since the dimensionality of the fingerprint is equal to the number of pixels, the fingerprint is unique for each camera, and the probability of two cameras sharing similar fingerprints is extremely small. The fingerprint is also stable over time. All these properties make it an excellent quantity suitable for many forensic tasks, such as device identification, device linking, and tampering detection.

The tutorial describes methods for estimating the fingerprint from images taken by the camera and methods for fingerprint detection. The estimator is derived using the maximum likelihood principle from a simplified sensor output model. The model is then used to formulate fingerprint detection as a two-channel hypothesis testing problem for which the generalized likelihood detector is derived. Due to its complexity, the generalized likelihood ratio test (GLRT) detector is replaced with a simplified but substantially faster detector computable using fast Fourier transform.

The performance of the introduced forensic methods is briefly demonstrated with real images. Throughout the text, references to previously published articles guide the interested reader to more detailed technical information

For completeness, we note that there exist approaches combining sensor noise detection with machine-learning classification [14]–[16]. References [14,] [17], and [18] extend the sensor-based forensic methods to scanners. An older version of this forensic method was tested for cell phone cameras in [16] and in [19], where the authors show that combining sensor-based forensic methods with methods that identify camera brand can decrease false alarms. The improvement reported in [19], however, is unlikely to hold for the newer version of the sensor noise forensic method presented in this tutorial, as the results appear to be heavily influenced by uncorrected effects discussed under "Sensor Fingerprint Estimation" in the section of the same name. The problem of pairing of a large number of images was studied in [20] using an ad hoc approach. Anisotropy of image noise for classification of images into scans, digital camera images, and computer art was discussed in [21].

ACKNOWLEDGMENT

The work on this article was supported by Air Force Office of Scientific Research grant FA9550-06-1-0046. 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 the Air Force Office of Scientific Research or of the U.S. government.

AUTHOR

Jessica Fridrich (fridrich@binghamton.edu) is a professor of electrical and computer engineering at Binghamton University (SUNY). She received her Ph.D. in systems science from Binghamton University in 1995 and her M.S. in applied mathematics from Czech Technical University in Prague in 1987. Her main interests are in steganography, steganalysis, digital watermarking, and digital image forensics. Since 1995, she received 18 research grants; most for projects on data embedding and steganalysis that lead to more than 80 papers and seven U.S. patents. She is a Member of the IEEE and ACM.

REFERENCES

[1] J. R. Janesick, "Scientific Charge-Coupled Devices," in *Monograph,* vol. PM83. Bellingham, WA: SPIE, 2001.

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

[3] S. M. Kay, "Fundamentals of Statistical Signal Processing," in *Estimation Theory,* vol. 1. Englewood Cliffs, NJ: Prentice-Hall, 1998.

[4] S. M. Kay, "Fundamentals of Statistical Signal Processing," in *Detection Theory*, vol. 2. Englewood Cliffs, NJ: Prentice-Hall, 1998.

[5] A. El Gamal, B. Fowler, H. Min, and X. Liu, "Modeling and estimation of FPN components in CMOS image sensors," in *Proc. SPIE, Solid State Sensor Arrays: Development and Applications II*, San Jose, CA, Jan. 1998, vol. 3301-20, pp. 168–177.

[6] T. Filler, J. Fridrich, and M. Goljan, "Using sensor pattern noise for camera model identification," in *Proc. IEEE ICIP 08*, San Diego, CA, Sept. 2008.

[7] C. R. Holt, "Two-channel detectors for arbitrary linear channel distortion," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, no. 3, pp. 267–273, Mar. 1987.

[8] B. V. K. Vijaya Kuma and L. Hassebrook, "Performance measures for correlation filters," *Appl. Opt.*, vol. 29, no. 20, pp. 2997–3006, 1990.

[9] M. Goljan and J. Fridrich, "Camera identification from cropped and scaled images," in *Proc. SPIE Electronic Imaging, Forensics, Security, Steganography, and Watermarking of Multimedia Contents X*, San Jose, CA, Jan. 28–30, 2008, vol. 6819, pp. 0E-1–0E-13.

[10] M. Goljan and J. Fridrich, "Camera identification from printed images," in *Proc. SPIE Electronic Imaging, Forensics, Security, Steganography, and Watermarking of Multimedia Contents X*, San Jose, CA, Jan. 28–30, 2008, vol. 6819, pp. OI-1–OI-12.

[11] M. Goljan, Mo Chen, and J. Fridrich, "Identifying common source digital camera from image pairs," in *Proc. IEEE ICIP 07*, San Antonio, TX, 2007.

[12] M. Chen, J. Fridrich, and M. Goljan, "Source digital camcorder identification using ccd photo response non-uniformity," in *Proc. SPIE Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, San Jose, CA, Jan. 28–Feb. 1, 2007, vol. 6505, pp. 1G–1H.

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

[14] H. Gou, A. Swaminathan, and M. Wu, "Robust scanner identification based on noise features," in *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, San Jose, CA, Jan. 29–Feb. 1, 2007, vol. 6505, pp. 0S–0T.

[15] N. Khanna, A. K. Mikkilineni, G. T. C. Chiu, J. P. Allebach, and E. J. *Delp*, III, "Forensic classification of imaging sensor types," in *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, San Jose, CA, Jan. 29–Feb. 1, 2007, vol. 6505, pp. 0U–0V.

[16] B. Sankur, O. Celiktutan, and I. Avcibas, "Blind identification of cell phone cameras," in *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, San Jose, CA, Jan. 29–Feb. 1, 2007, vol. 6505, pp. 1H–1I.

[17] T. Gloe, E. Franz, and A. Winkler, "Forensics for flatbed scanners," in *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, San Jose, CA, Jan. 29–Feb. 1, 2007, vol. 6505, pp. 1I–1J.

[18] N. Khanna, A. K. Mikkilineni, G. T. C. Chiu, J. P. Allebach, and E. J. Delp, III, "*Scanner identification using sensor pattern noise," in Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, San Jose, CA, Jan. 29–Feb. 1, 2007, vol. 6505, pp. 1K–1L.

[19] Y. Sutcu, S. Bayram, H. T. Sencar, and N. Memon, "Improvements on sensor noise based source camera identification," in *Proc. IEEE Int. Conf. Multimedia and Expo*, July 2007, pp. 24–27.

[20] G. J. Bloy, "Blind camera fingerprinting and image clustering," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 30, no. 3, pp. 532–534, Mar. 2008.

[21] N. Khanna, G. T.-C. Chiu, J. P. Allebach, and E. J. Delp, "Forensic techniques for classifying scanner, computer generated and digital camera images," in *Proc. IEEE ICASSP*, Mar. 31–Apr. 4, 2008, pp. 1653–1656.

[22] M. K. Mihcak, I. Kozintsev, and K. Ramchandran, "Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Phoenix, AZ, Mar. 1999, vol. 6, pp. 3253–3256. [SP]