Chapter 1
Sensor Defects in Digital Image Forensic

Jessica Fridrich

Just as human fingerprints or skin blemishes can be used for forensic purposes, imperfections of digital imaging sensors can serve as unique identifiers in numerous forensic applications, such as matching an image to a specific camera, revealing malicious image manipulation and processing, and determining an approximate age of a digital photograph. There exist several different types of defects that are of interest to the forensic analyst caused by imperfections in manufacturing, physical processes occurring inside the camera, and by environmental factors. This chapter begins with analyzing pixel defects, while pointing out their forensic potential. Then, specific problems are formulated as tasks involving detection or matching of defects and noise patterns. Practical algorithms for these tasks are developed within the framework of parameter estimation and signal detection theory. The performance of the algorithms is demonstrated on real-world examples.

1.1 Introduction

In the heart of every electronic device capable of taking digital pictures is an imaging sensor. There exist two types of sensors – CCD (Charge-Coupled Device) and CMOS (Complementary Metal-Oxide Semiconductor). Both sensors consist of a large number of photo detectors commonly called pixels. Pixels are made of silicon and capture light by converting photons into electrons using the photoelectric effect [30, 27]. The charge accumulated at every pixel is transferred out of the sensor, amplified, and then run through an AD converter that converts it to a digital signal. The digitized signal is further processed before the data is stored as an electronic file (JPEG, TIFF, etc.) on the camera storage device. The pixels are several microns across and have a rectangular shape. In theory, the amount of electrons (charge) outputted by a pixel should depend solely on the intensity of the incident light. In reality,

however, there are many factors that introduce both systematic and random deviations. It is exactly these fluctuations that find important applications in forensic analysis.

We will be interested primarily in *systematic* variations in pixel response that manifest themselves in a consistent manner in all images because random fluctuations that change independently from scene to scene would not be particularly useful. In the next section, we describe several types of such systematic sensor defects and how they affect the output of a sensor. By doing so, we arrive at a model of sensor output that will be useful later for deriving estimators of parameters that describe the defects and for constructing defect detectors. In Section 1.3, we introduce the concept of sensor fingerprint and derive a procedure using which the fingerprint can be estimated from images taken by the camera. The tasks of camera identification, device linking, and forgery detection can be approached by testing the presence of sensor fingerprint in images. This is the subject of Sections 1.4, 1.4.1, and 1.4.4. The methods are tested on real images in Section 1.5. In Section 1.6, we develop methods that can address counter-forensic activities of the type when an adversary estimates a camera fingerprint and then adds it to an image from a different camera to frame an innocent victim. The presence or absence of defects can also provide temporal information for estimating an approximate age of digital photographs. This so-called temporal forensics is the subject of Section 1.7. The chapter is summarized in Section 1.8.

1.1.1 Notation

Everywhere in this chapter, boldface font will denote vectors (or matrices) of length specified in the text, e.g., $\mathbf{X}$ and $\mathbf{Y}$ are vectors of length $m \times n$ 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, etc., are *elementwise*. The Euclidean dot product of vectors is denoted as $\mathbf{X} \cdot \mathbf{Y}$ with $\|\mathbf{X}\| = \sqrt{\mathbf{X} \cdot \mathbf{X}}$ being the L_2 (Euclidean) norm of $\mathbf{X}$. Denoting the sample mean with a bar, the normalized correlation is

$$corr(\mathbf{X}, \mathbf{Y}) = \frac{(\mathbf{X} - \overline{\mathbf{X}}) \cdot (\mathbf{Y} - \overline{\mathbf{Y}})}{\left\|\mathbf{X} - \overline{\mathbf{X}}\right\| \left\|\mathbf{Y} - \overline{\mathbf{Y}}\right\|}. \tag{1.1}$$

For a logical statement P, we also make use of the Iverson bracket defined as $[P] = 1$ when P is true and $[P] = 0$ when P is false.

1.2 Imaging sensors and their defects

In this section, we explain two types of systematic defects – the photo-response non-uniformity and dark current – that are useful for several important forensic tasks, including camera identification and forgery detection. Then, we formulate a model of pixel output that will be used in the rest of this chapter to build all necessary mathematical tools for the forensic analyst.

1.2.1 Photo-response non-uniformity

The charge generated in a pixel depends on the physical dimensions of the pixel photosensitive area and on the homogeneity of silicon. The pixels' physical dimensions slightly vary 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 variations in quantum efficiency among pixels can be captured with a matrix $\mathbf{K} \in \mathbb{R}^{m \times n}$ of the same dimensions as the sensor. When an imaging sensor is illuminated with light intensity $\mathbf{I} \in \mathbb{R}^{m \times n}$, in the absence of other noise sources or imperfections, the sensor would register a noisy scene $\mathbf{I} + \mathbf{IK}$ instead. (We remind that the product $\mathbf{IK}$ is an element-wise product of matrices) The term $\mathbf{IK}$ is usually referred to as the photo-response non-uniformity or PRNU. A large value of $\mathbf{K}$ leads to a point defect called "pixel with abnormal sensitivity."

One can say that the scene $\mathbf{I}$ is overlaid with a "noise pattern" $\mathbf{IK}$, which is essentially the matrix $\mathbf{K}$ modulated by the scene $\mathbf{I}$. Note that the energy of this noise pattern depends on the light intensity – it is larger for bright images and smaller in pictures of mostly dark scenes.

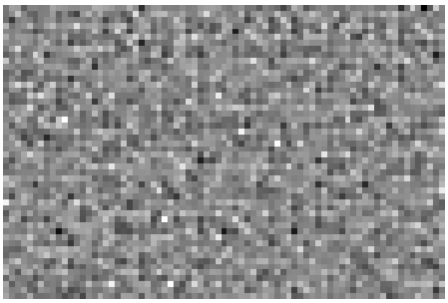

Fig. 1.1 Closeup of the PRNU factor $\mathbf{K}$ enhanced for visualization.

Fig. 1.1 shows a magnified portion of a the PRNU factor $\mathbf{K}$ from a four-megapixel camera Canon G2. Bright dots correspond to pixels that consis-

tently generate more electrons, while dark dots mark pixels whose response is consistently lower. To give the reader a sense of how weak the PRNU signal **IK** typically is, for a picture of a uniform background **I** with average grayscale in the middle of the dynamic range (grayscale $\mathbf{I} = 128$ for an 8-bit image), the average energy of $\mathbf{I}(i)\mathbf{K}(i)$ over all pixels i is 0.5, which can also be formulated as SNR of 51 dB. The energy of the PRNU strongly varies among camera models.

In Section 1.3, it is shown how the PRNU factor **K** can be used as a sensor "fingerprint" for a variety of forensic tasks.

1.2.2 Dark current

Even when a pixel is not exposed to light during picture taking, it contains a small number of free electrons due to thermal effects. Their number increases with temperature and exposure [26, 30, 28]. It is also affected by ISO setting. In the absence of all other defects, the pixel's output is $\mathbf{I}+\tau\mathbf{D}+\mathbf{c}$, where $\tau\mathbf{D}$ is called the dark current and **c** the offset. Here, $\tau \geq 0$ is a multiplicative factor whose value is determined by the temperature, exposure, and ISO (higher ISO leads to a larger value of τ) and $\mathbf{D}, \mathbf{c} \in \mathbb{R}^{m \times n}$ are matrices. In images taken with a short exposure (e.g., 1/60th of a second or shorter), the dark current is usually very weak. However, it may start dominating the sensor output for dark scenes ($\mathbf{I} \approx 0$) when τ becomes large (large ISO and/or temperature and/or long exposure). An extremely high value of **D** produces the most common point defect called a hot pixel. A high value of the offset **c** leads to another defect type commonly recognized as a stuck pixel. Both defects were proposed for forensic tasks in 1999 by Kurosawa [35], who demonstrated that as long as a video clip contained some dark frames, hot/stuck pixels can be used to uniquely identify digital video cameras.

Hot/stuck pixels occur randomly and uniformly on the sensor independently of each other, which makes them useful for determining an approximate age of digital photographs (see Section 1.7.2).

1.2.3 Pixel output model

Considering the impact of all the defects discussed so far, we arrive at the following model for the raw output of a sensor:

$$\mathbf{Y} = \mathbf{I} + \mathbf{IK} + \tau\mathbf{D} + \mathbf{c} + \boldsymbol{\Theta}. \tag{1.2}$$

We remind that τ is a scalar multiplicative factor whose value is determined by exposure, temperature, and ISO settings. The matrix **c** is the matrix of

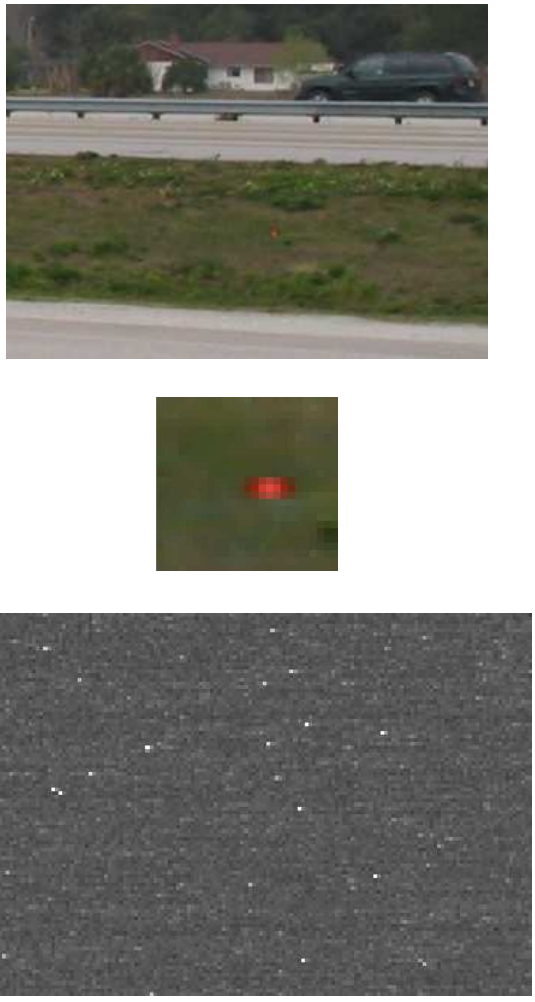

Fig. 1.2 Top: A stuck pixel in an image and its close up. The pixel happens to be red because it has a red color filter in front of it. Bottom: An example of a dark frame. Notice that there appears to be a degree of "hotness" among the pixels.

offsets and dark current factor $\mathbf{D}$, is a noise-like signal due to leakage of electrons into pixels' electron wells (Fig. 1.2 bottom). Finally, the reader recognizes $\mathbf{K}$ as the PRNU factor. The modeling noise $\boldsymbol{\Theta}$ is a collection of all other noise sources, which are mostly random in nature and thus difficult to use for forensic purposes (readout noise, shot noise, also known as the photonic noise, quantization noise, etc.).

It should be stressed that the defects represented by matrices $\mathbf{K}$, $\mathbf{D}$, and $\mathbf{c}$ usually represent quite small deviations with the exception of spike pixel

defects, such as hot or stuck pixels. The three matrices could be estimated from multiple images taken by the camera.

To improve the signal to noise ratio between the signal of interest (defect) and the observable $\mathbf{Y}$, we usually work with the noise residual $\mathbf{W} = \mathbf{Y} - F(\mathbf{Y})$, obtained using a denoising filter F. In particular,

$$\begin{aligned}\mathbf{W} &= \mathbf{Y} - F(\mathbf{Y}) \\ &= \mathbf{IK} + \tau\mathbf{D} + \mathbf{c} + \mathbf{I} - F(\mathbf{Y}) + \boldsymbol{\Theta} \\ &= \mathbf{IK} + \tau\mathbf{D} + \mathbf{c} + \boldsymbol{\Xi},\end{aligned} \tag{1.3}$$

where $\boldsymbol{\Xi}$ stands for the sum of the modeling noise and the remnant of the content $\mathbf{I} - F(\mathbf{Y})$ present due to the inability of the denoising filter to separate content from noise. The term $\mathbf{I} - F(\mathbf{Y})$ is especially large in textured regions and around edges.

A variety of filters could be used in practice for this task. In this chapter, we will use the wavelet filter [43] designed to suppress a non-stationary Gaussian noise and a 3×3 median filter.

1.3 Sensor fingerprint

The following five properties of PRNU represented with matrix $\mathbf{K}$ are the main reason why it was proposed to play the role of a sensor fingerprint.

1. **Dimensionality**. The matrix $\mathbf{K}$ appears random, which gives it a large information content and makes it unique to each sensor. The probability of two sensors having similar fingerprints is extremely low. Even two cameras of the same model have statistically independent fingerprints.
2. **Universality**. All imaging sensors exhibit PRNU.
3. **Generality**. The PRNU component $\mathbf{IK}$ is present in every picture independently of the camera optics, camera settings, or scene content, with the exception of completely dark images, where $\mathbf{I} \approx 0$.
4. **Stability**. The factor $\mathbf{K}$ is stable in time and under wide range of environmental conditions (temperature, humidity).
5. **Robustness**. The PRNU component $\mathbf{IK}$ survives lossy compression, filtering, gamma correction, and many other typical processing.

The elements of the matrix $\mathbf{K}$ are well-modeled as independent and identically distributed (iid) realizations of a Gaussian random variable. By establishing the presence of the PRNU signal $\mathbf{IK}$ in an image, one can prove with a high level of certainty that the image was obtained by a *specific* camera whose fingerprint is *known*. This application is called sensor (camera) identification. Expanding this application further, by detecting the presence of the signal $\mathbf{IK}$ in individual image regions, one can reveal that certain regions were replaced or tampered with; this application is known as integrity verification or forgery

detection. Alternatively, by proving that two images share a common signal, it is possible to establish that these two images came from the same device even when its fingerprint is not available. This is called device linking. Finally, the PRNU signal **IK** can be used as a template to recover geometrical processing the image has been subjected to, such as cropping, resizing, or rotation.

In the sections below, we explain a procedure for estimating the sensor fingerprint as well as methods for its detection in images to address the problem of camera identification, device linking, fingerprint matching, and forgery detection. The performance of these methods is evaluated on real imagery in Section 1.5.

1.3.1 Fingerprint estimation

The problem of estimating the fingerprint $\mathbf{K}$ can be approached using standard techniques from parameter estimation. The estimator will depend on the model of sensor output. Ideally, the best estimator should be tailored to the specific camera make and model, while taking into account its processing pipeline. There is, however, substantial strength in approaching the estimation using a simplified model that is universally valid for virtually all camera makes and models. This avenue is taken in this chapter.

Starting with the model of the noise residual $\mathbf{W} = \mathbf{Y} - F(\mathbf{Y})$ (1.3), we simplify it by including the dark current and the offset into the noise term:

$$\mathbf{W} = \mathbf{IK} + \boldsymbol{\Xi}. \tag{1.4}$$

It is easier to estimate the PRNU term from $\mathbf{W}$ than from $\mathbf{Y}$ because the image content is greatly suppressed in $\mathbf{W}$.

Let us assume that we have a database of $d \geq 1$ images, $\mathbf{Y}_1, \ldots, \mathbf{Y}_d$, obtained by the camera whose fingerprint we wish to estimate. For each pixel i, we model the sequence $\boldsymbol{\Xi}_1(i), \boldsymbol{\Xi}_2(i), \ldots, \boldsymbol{\Xi}_d(i)$ as a white Gaussian noise (WGN) with variance $\sigma^2(i)$. The noise term is technically not independent of the PRNU signal **IK** due to the content leftover $\mathbf{I} - F(\mathbf{Y})$ in $\boldsymbol{\Xi}$. However, because the energy of this term is small compared to **IK**, the assumption that $\boldsymbol{\Xi}$ is independent of **IK** is reasonable.

From (1.4), we can write for each $k = 1, \ldots, d$ in a matrix form:

$$\frac{\mathbf{W}_k}{\mathbf{I}_k} = \mathbf{K} + \frac{\boldsymbol{\Xi}_k}{\mathbf{I}_k}. \tag{1.5}$$

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

$$L(\mathbf{K}) = -\frac{d}{2}\sum_{k=1}^{d}\log(2\pi\sigma^2/\mathbf{I}_k^2) - \sum_{k=1}^{d}\left(\frac{\mathbf{W}_k/\mathbf{I}_k - \mathbf{K}}{2\sigma^2/\mathbf{I}_k^2}\right)^2. \tag{1.6}$$

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

$$\hat{\mathbf{K}} = \frac{\sum_{k=1}^{d} \mathbf{I}_k \mathbf{W}_k}{\sum_{k=1}^{d} \mathbf{I}_k^2}. \tag{1.7}$$

We remind that all operations in (1.7) are elementwise. To be able to use this estimator in practice, we can simply set $\mathbf{I}_k = \mathbf{Y}_k$ or $\mathbf{I}_k = F(\mathbf{Y}_k)$ because the PRNU term is weak.

We also define the quality of a fingerprint estimate as

$$q = corr(\mathbf{K}, \hat{\mathbf{K}}). \tag{1.8}$$

The Cramer-Rao Lower Bound (CRLB) [31] gives us the bound on the variance of $\hat{\mathbf{K}}$

$$\frac{\partial^2 L(\mathbf{K})}{\partial \mathbf{K}^2} = -\frac{\sum_{k=1}^{d} \mathbf{I}_k^2}{\sigma^2}, \tag{1.9}$$

which implies

$$Var(\hat{\mathbf{K}}) \geq \left(-E\left(\frac{\partial^2 L(\mathbf{K})}{\partial \mathbf{K}^2} \right) \right)^{-1} = \frac{\sigma^2}{\sum_{k=1}^{d} \mathbf{I}_k^2}. \tag{1.10}$$

Because the sensor model is linear, the CRLB tells us that the maximum likelihood estimator is minimum variance unbiased and its variance is proportional to d. Therefore, the best images for estimating the fingerprint are those with high luminance (but not saturated) and small σ^2 (images with a smooth content). If the camera under investigation is available to the analyst, unsaturated out-of-focus images of bright cloudy sky would be the best. In practice, good estimates of the fingerprint may be obtained from as few as 20 natural images depending on the camera. If sky images are used instead of natural images, only approximately one half of them would suffice to obtain an estimate with a comparable accuracy.

1.3.1.1 Fingerprint post-processing

The fingerprint 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 [15], and sensor design. While the PRNU is unique to the sensor, the above Non-Unique Artifacts (NUAs) 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, since most of these artifacts are due to demosaicking algorithms that depend on the Color Filter Array (CFA) and are periodic in nature, they can be

Algorithm 1 The procedure of zero-meaning removes NUAs from the fingerprint estimated using (1.7).

$r_i = 1/n \sum_{j=1}^{n} \hat{\mathbf{K}}^T(i,j)$ \\ compute row averages
for $i = 1$ to m \\ zero-mean rows
  $\hat{\mathbf{K}}^T(i,j) \leftarrow \hat{\mathbf{K}}^T(i,j) - r_i$ for $j = 1$ to n
end
$c_j = 1/m \sum_{i=1}^{m} \hat{\mathbf{K}}^T(i,j)$ \\ compute column averages
for $j = 1$ to n \\ zero-mean rows
  $\hat{\mathbf{K}}^T(i,j) \leftarrow \hat{\mathbf{K}}^T(i,j) - c_j$ for $i = 1$ to m
end

removed by zero-meaning the rows and columns of separately for each pixel type as defined by the CFA. We explain the procedure on the example of the Bayer CFA.

Assuming $\mathbf{I}$ has $m \times n$ pixels, for the Bayer CFA there are four types of pixels forming four interleaved submatrices $\hat{\mathbf{K}}^T$, $T \in \{R, G1, G2, B\}$. where $\hat{\mathbf{K}}^T$ is of dimension $(m/2) \times (n/2)$. The operation of zero-meaning is described using Algorithm 1.

Reassembling the four submatrices into one $m \times n$ matrix again, the magnitude of the final fingerprint estimate is further processed in the DFT domain using a Wiener filter with noise variance σ^2, $W(., \sigma^2)$, to further suppress any remaining NUAs, such as non-periodic artifacts [8] (all operations are again elementwise):

$$\mathbf{F} = \mathcal{F}(\hat{\mathbf{K}}), \quad \hat{\mathbf{K}} \leftarrow \text{Real}\left[\mathcal{F}^{-1}\left(\mathbf{F} \cdot \frac{|\mathbf{F}| - W(|\mathbf{F}|, \sigma^2)}{|\mathbf{F}|}\right)\right], \tag{1.11}$$

where $\mathcal{F}$ is the orthonormal Fourier transform and $\sigma^2 = \frac{1}{mn} \sum_{i,j} \hat{\mathbf{K}}^2(i,j)$.

The difference between the original estimate (1.7) and the post-processed $\hat{\mathbf{K}}$ is called the linear pattern (see Fig. 1.3) and it is a useful forensic entity by itself – it can be used to classify a camera fingerprint to a camera model or brand. The reader is referred to [14] for more details.

For color images, the PRNU factor can be estimated for each color channel separately, obtaining thus three fingerprints of the same dimensions $\hat{\mathbf{K}}_R$, $\hat{\mathbf{K}}_G$, and $\hat{\mathbf{K}}_B$. As these three fingerprints are highly correlated due to in-camera processing, such as demosaicking or color interpolation, an analyst may choose to work with a single fingerprint obtained by converting the three color fingerprints using the usual conversion from RGB to grayscale:

$$\hat{\mathbf{K}} = 0.3\hat{\mathbf{K}}_R + 0.6\hat{\mathbf{K}}_G + 0.1\hat{\mathbf{K}}_B. \tag{1.12}$$

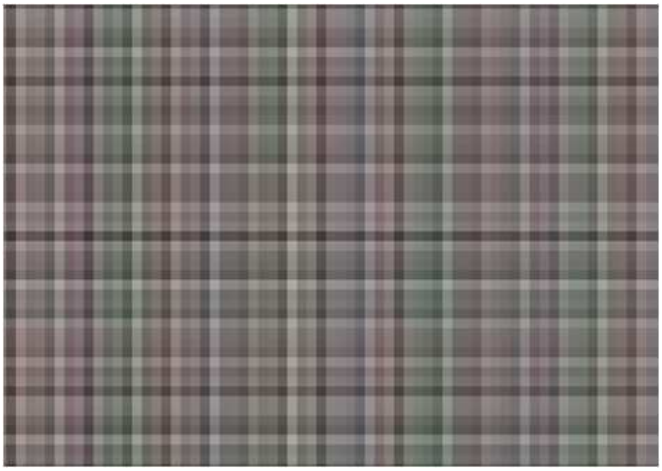

Fig. 1.3 The NUAs in the form of a linear pattern for a Canon S40 camera.

1.4 Digital forensics using sensor fingerprint

Historically the first application of sensor fingerprint was camera identification [40]. The goal is to determine whether an image under investigation was taken with a specific camera whose fingerprint is available. This does not necessarily mean that the camera needs to be physically available to the analyst because the fingerprint can be estimated from images that provably came from the camera. The identification is achieved by testing whether the noise residual of the image under investigation contains traces of the camera fingerprint.

We formulate the hypothesis testing problem for camera identification in a setting that is general enough to conveniently cover the remaining forensic tasks – device linking and fingerprint matching. In device linking, two images are tested if they came from the same camera (the camera itself is not 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).

1.4.1 Device identification

We consider a more general scenario in which the image under investigation has possibly undergone a geometrical transformation, such as scaling, rotation, or cropping. For simplicity of explanation, we will also assume that

before applying any geometrical transformation the image was in grayscale represented with an $m \times n$ matrix $\mathbf{I}(i,j)$. The geometrical transformation is a complication because the image and the sensor fingerprint are no longer synchronized.

Let us denote as $\mathbf{u}$ the (unknown) vector of parameters describing the geometrical transformation, which we denote $T_{\mathbf{u}}$. For example, $\mathbf{u}$ could be a scaling ratio or a two-dimensional vector consisting of the scaling parameter and an unknown angle of rotation. In device identification, we wish to determine whether or not the transformed image $\mathbf{Z}$ 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 with the fingerprint rather than matching $\mathbf{Z}$ with a transformed version of $\hat{\mathbf{K}}$.

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

$$\begin{aligned} H_0: &\quad \mathbf{K}_1 \neq \mathbf{K}_2, \\ H_1: &\quad \mathbf{K}_1 = \mathbf{K}_2, \end{aligned} \tag{1.13}$$

where

$$\begin{aligned} \mathbf{W}_1 &= \mathbf{I}_1\mathbf{K}_1 + \boldsymbol{\Xi}_1, \\ T_{\mathbf{u}}^{-1}(\mathbf{W}_2) &= T_{\mathbf{u}}^{-1}(\mathbf{Z})\mathbf{K}_2 + \boldsymbol{\Xi}_2. \end{aligned} \tag{1.14}$$

Here, all signals are observed with the exception of the noise terms, and the fingerprints $\mathbf{K}_1$ and $\mathbf{K}_2$. In particular, for the device identification problem, we have $\mathbf{I}_1 = 1$, $\mathbf{W}_1 = \hat{\mathbf{K}}$ estimated in the previous section, and $\boldsymbol{\Xi}_1$ 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 $\boldsymbol{\Xi}_2$ is a noise term. In general, $\mathbf{u}$ is an unknown nuisance parameter. Note that since $T_{\mathbf{u}}^{-1}(\mathbf{W}_2)$ and $\mathbf{W}_1$ may have different dimensions, the formulation (1.13)–(1.14) involves an unknown spatial shift between both signals, $\mathbf{s}$.

Modeling the noise terms and as white Gaussian noise with known variances σ_1^2, σ_2^2, the generalized likelihood ratio test [32] for this two-channel problem was derived in [29]. The test statistic t

$$t = \max_{\mathbf{u},\mathbf{s}} \{E_1(\mathbf{u},\mathbf{s}) + E_2(\mathbf{u},\mathbf{s}) + C(\mathbf{u},\mathbf{s})\}, \tag{1.15}$$

is a sum of three terms: two energy-like quantities and a cross-correlation term:

$$E_1(\mathbf{u},\mathbf{s}) = \sum_{i,j} \frac{\mathbf{I}_1^2(i,j)(\mathbf{W}_1(i+s_1,j+s_2))^2}{\sigma_1^2\mathbf{I}_1^2(i,j) + \sigma_1^4\sigma_2^{-2}\left(T_{\mathbf{u}}^{-1}(\mathbf{Z})(i+s_1,j+s_2)\right)^2}, \tag{1.16}$$

$$E_2(\mathbf{u},\mathbf{s}) = \sum_{i,j} \frac{\left(T_{\mathbf{u}}^{-1}(\mathbf{Z})(i+s_1,j+s_2)\right)^2\left(T_{\mathbf{u}}^{-1}(\mathbf{W}_2)(i+s_1,j+s_2)\right)^2}{\sigma_2^2\left(T_{\mathbf{u}}^{-1}(\mathbf{Z})(i+s_1,j+s_2)\right)^2 + \sigma_2^4\sigma_1^{-2}\mathbf{I}_1^2(i,j)}, \tag{1.17}$$

$$C(\mathbf{u},\mathbf{s}) = \sum_{i,j} \frac{\mathbf{I}_1\mathbf{W}_1(i,j)(T_{\mathbf{u}}^{-1}(\mathbf{Z})(i+s_1,j+s_2))(T_{\mathbf{u}}^{-1}(\mathbf{W}_2)(i+s_1,j+s_2))}{\sigma_2^2\mathbf{I}_1^2(i,j) + \sigma_1^2\left(T_{\mathbf{u}}^{-1}(\mathbf{Z})(i+s_1,j+s_2)\right)^2}. \tag{1.18}$$

The complexity of evaluating these three expressions is proportional to the square of the number of pixels, $(mn)^2$, which makes this detector unusable in practice. Thus, we simplify this detector and give it the form of a Normalized Cross-Correlation (NCC) that can be evaluated using the fast Fourier transform. Under H_1, the maximum in (1.15) is mainly due to the contribution of the cross-correlation term, $C(\mathbf{u},\mathbf{s})$, that 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 $\mathbf{u}$,

$$ncc(s_1,s_2,\mathbf{u}) = \frac{\sum_{i,j=1}^{m,n}(\mathbf{X}(i,j)-\overline{\mathbf{X}})(\mathbf{Y}(i+s_1,j+s_2)-\overline{\mathbf{Y}})}{\left\|\mathbf{X}-\overline{\mathbf{X}}\right\|\left\|\mathbf{Y}-\overline{\mathbf{Y}}\right\|}, \tag{1.19}$$

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

$$\mathbf{X} = \frac{\mathbf{I}_1\mathbf{W}_1}{\sqrt{\sigma_2^2\mathbf{I}_1^2 + \sigma_1^2\left(T_{\mathbf{u}}^{-1}(\mathbf{Z})\right)^2}}, \quad \mathbf{Y} = \frac{T_{\mathbf{u}}^{-1}(\mathbf{Z})T_{\mathbf{u}}^{-1}(\mathbf{W}_2)}{\sqrt{\sigma_2^2\mathbf{I}_1^2 + \sigma_1^2\left(T_{\mathbf{u}}^{-1}(\mathbf{Z})\right)^2}}. \tag{1.20}$$

A more stable detection statistics, whose meaning will become apparent from error analysis later in this section, that we strongly advocate to use for all camera identification tasks, is the Peak to Correlation Energy measure (PCE):

$$PCE(\mathbf{u}) = \frac{ncc(\mathbf{s}_{\text{peak}},\mathbf{u})^2}{\frac{1}{mn-|\mathcal{N}|}\sum_{\mathbf{s}\notin\mathcal{N}} ncc(\mathbf{s},\mathbf{u})^2}, \tag{1.21}$$

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

For device identification from a single image, the fingerprint estimation noise $\boldsymbol{\Xi}_1$ is much weaker compared with $\boldsymbol{\Xi}_2$ – the noise residual of the image under investigation. Thus, $\sigma_1^2 = Var(\boldsymbol{\Xi}_1) = Var(\boldsymbol{\Xi}_2) = \sigma_2^2$ and (1.19) simplifies to a normalized cross-correlation between

$$\mathbf{X} = \mathbf{W}_1 = \hat{\mathbf{K}} \quad \text{and} \quad \mathbf{Y} = T_{\mathbf{u}}^{-1}(\mathbf{Z})T_{\mathbf{u}}^{-1}(\mathbf{W}_2). \tag{1.22}$$

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 statistic is noise-like 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 illustrative example, we next provide additional details how one can carry out the search for the simpler case when the image is known to have been subjected only to a combination of scaling and cropping, in which case $\mathbf{u} = r$ is an unknown scaling ratio. The reader is advised to consult [21] for more details.

Fig. 1.4 The original image and its cropped and scaled version. The scaling ratio was $r = 0.51$. The light gray rectangle shows the original image size, while the dark gray frame shows the size after cropping but before resizing.

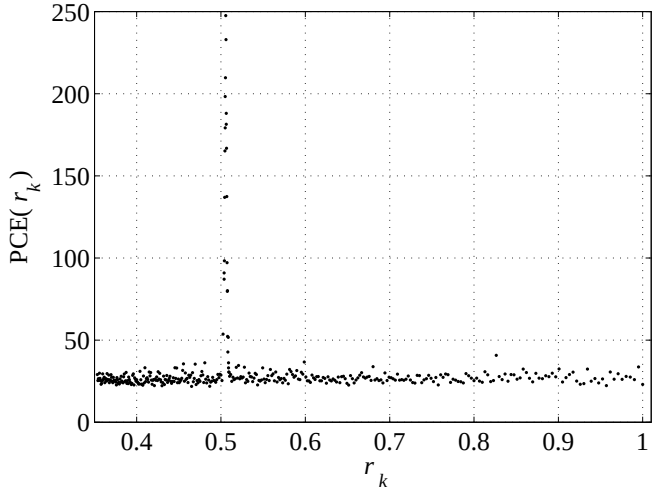

Fig. 1.5 Search for the scaling ratio. The peak in $PCE(r_k)$ was detected around the correct ratio of 0.51.

1.4.1.1 Identification from scaled images

Assuming the image under investigation $\mathbf{Z}$ has dimensions $M \times N$, we search for the scaling parameter at discrete values $r_k \leq 1$, $k = 0, 1, \ldots, R$, from $r_0 = 1$ (no scaling, just cropping) down to $r_R = \max\{M/m, N/n\} < 1$:

$$r_k = \frac{1}{1 + 0.005k}, \quad k = 0, 1, 2, \ldots. \tag{1.23}$$

This particular form of the search grid is fine enough to not miss the correct scaling ratio but not too dense as that would slow down the search. After the ratio is found, it is possible to further refine the estimate by a secondary search around a small neighborhood of the ratio just found.

For a fixed scaling parameter r_k, the cross-correlation (1.19) does not have to be computed for all shifts $\mathbf{s}$ but only for those that move the upsampled image $T_{r_k}^{-1}(\mathbf{Z})$ within the dimensions of $\hat{\mathbf{K}}$ because only such shifts can be generated by cropping. Given that the dimensions of the upsampled image are $M/r_k \times N/r_k$, we have the following range for the spatial shift $\mathbf{s} = (s_1, s_2)$:

$$0 \leq s_1 \leq m - M/r_k \quad \text{and} \quad 0 \leq s_2 \leq n - N/r_k. \tag{1.24}$$

The peak of the two-dimensional NCC across all spatial shifts $\mathbf{s}$ is evaluated for each r_k using $PCE(r_k)$. If $\max_k PCE(r_k) > \tau$, we decide H_1 (camera and image are positively matched). Moreover, the value of the scaling parameter at which the PCE attains its maximum determines the scaling ratio r_{peak}. The location of the peak, $\mathbf{s}_{\text{peak}}$, in the normalized cross-correlation determines

the cropping parameters. Thus, as a by-product of this algorithm, we can determine the processing history of $\mathbf{Z}$ (see Fig. 1.4 bottom). The fingerprint is essentially playing the role of a synchronizing template. It can also be used for reverse-engineering in-camera processing, such as digital zoom [21] or blind estimation of focal length at which the image was taken for cameras that correct for lens distortion inside the camera [22].

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_k, the values of the normalized cross-correlation $ncc(\mathbf{s}, r_k)$ as a function of $\mathbf{s}$ are well-modeled [21] as white Gaussian noise $\xi_k \sim N(0, \sigma_k^2)$ with variance that may depend on k. Estimating the variance of the Gaussian model using the sample variance of $ncc(\mathbf{s}, r_k)$ over $\mathbf{s}$ after excluding a small central region $\mathcal{N}$ surrounding the peak,

$$\hat{\sigma}_k^2 = \frac{1}{mn - |\mathcal{N}|} \sum_{\mathbf{s} \notin \mathcal{N}} ncc(\mathbf{s}, r_k)^2. \tag{1.25}$$

We now calculate the probability p_k that the NCC would attain the peak value $ncc(\mathbf{s}_{\text{peak}}, r_{\text{peak}})$ or larger by chance:

$$\begin{aligned} p_k &= \int_{ncc(\mathbf{s}_{\text{peak}}, r_{\text{peak}})}^{\infty} \frac{1}{\sqrt{2\pi}\hat{\sigma}_k} \exp(x^2/2\hat{\sigma}_k^2) \mathrm{d}x, \\ &= \int_{\hat{\sigma}_{\text{peak}}\sqrt{PCE_{\text{peak}}}}^{\infty} \frac{1}{\sqrt{2\pi}\hat{\sigma}_k} \exp(x^2/2\hat{\sigma}_k^2) \mathrm{d}x, \\ &= Q\left(\frac{\hat{\sigma}_{\text{peak}}}{\hat{\sigma}_k} \sqrt{PCE_{\text{peak}}} \right), \end{aligned} \tag{1.26}$$

where $Q(x) = 1 - \Phi(x)$ with $\Phi(x)$ denoting the cumulative distribution function of a standard normal variable $N(0, 1)$ and $PCE_{\text{peak}} = PCE(r_{\text{peak}})$.

As explained above, during the search for the cropping vector $\mathbf{s}$, we need to search only in the range (1.24), which means that we are taking maximum over $l_k = (m - M/r_k + 1) \times (n - N/r_k + 1)$ samples of ξ_k. Thus, the probability that the maximum value of ξ_k would not exceed $ncc(\mathbf{s}_{\text{peak}}, r_{\text{peak}})$ is $(1 - p_k)^{l_k}$. After R steps in the search, the probability of false alarm is

$$P_{FA} = 1 - \prod_{k=1}^{R} (1 - p_k)^{l_k}. \tag{1.27}$$

Since we can stop the search after the PCE reaches a certain threshold, we have $r_k \leq r_{\text{peak}}$. Because $\hat{\sigma}_k$ is non-decreasing in k, $\hat{\sigma}_{\text{peak}}/\hat{\sigma}_k \geq 1$. Because $Q(x)$ is decreasing, we have $p_k \leq Q(\sqrt{PCE_{\text{peak}}})$. Thus, because $l_k \leq mn$, we obtain an upper bound on P_{FA}

$$P_{FA} \leq 1 - (1-p)^{l_{\max}}, \tag{1.28}$$

where $l_{\max} = \sum_{k=0}^{R-1} l_k$ is the maximal number of values of the parameters r and $\mathbf{s}$ over which the maximum of (1.15) could be taken. Equation (1.28), together with $p = Q(\sqrt{\tau})$, determines the threshold for PCE, $\tau = \tau(P_{FA}, M, N, m, n)$.

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 in Section 1.5 some experiments on real images. This algorithm can also be used with small modifications for device linking and fingerprint matching. A large-scale test of this methodology appears in [19].

Another extension of the identification algorithm to work with cameras that correct images for lens barrel/pincushion distortion on the fly appears in [22]. Such cameras are becoming very ubiquitous as manufacturers strive to offer customers a powerful optical zoom in inexpensive and compact cameras. Since the lens distortion correction is a geometrical transformation that depends on the focal length (zoom), there can be a desynchronization between the noise residual and the fingerprint if both were estimated at different focal lengths. Here, a search for the distortion parameter is again needed to resynchronize the signals.

1.4.2 Device linking

The detector derived in the previous section can be readily used with only a few changes for determining whether two images, $\mathbf{I}_1$ and $\mathbf{Z}$, were taken by the exact same camera [18], an application called device linking. Note that in this problem the camera or its fingerprint are not necessarily available.

The device linking problem corresponds exactly to the two-channel formulation (1.13) and (1.14) with the GLRT detector (1.15). Its faster, suboptimal version is the PCE (1.21) obtained from the maximum value of $ncc(\mathbf{s}_{\text{peak}}, r_{\text{peak}})$ over all $s_1, s_2, \mathbf{u}$ (see (1.19) and (1.20)). In contrast to the camera identification problem, the power of both noise terms, $\boldsymbol{\Xi}_1$ and $\boldsymbol{\Xi}_2$, is now comparable and needs to be estimated from observations. Fortunately, because the PRNU term $\mathbf{IK}$ is much weaker than the modeling noise $\boldsymbol{\Xi}$, reasonable estimates of the noise variances are simply $\hat{\sigma}_1^2 = Var(\mathbf{W}_1)$, $\hat{\sigma}_2^2 = Var(\mathbf{W}_2)$.

Unlike in the camera identification problem, the search for unknown scaling must now be enlarged to scalings $r > 1$ (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 Section 1.4.1.1. Due to space limitations we do not include experimental verification of the device linking algorithm. Instead, the reader is referred to [18].

1.4.3 Fingerprint matching

The last fingerprint matching scenario corresponds to the situation when we need to decide whether or not two estimates of potentially two different fingerprints are identical. This happens, for example, in video-clip linking because the fingerprint can be estimated from all frames forming the clip [10].

The detector derived in Section 1.4.1 applies to this scenario, as well. It can be further simplified because for matching fingerprints we have $\mathbf{I}_1 = \mathbf{Z} = 1$ and (1.19) becomes the normalized cross-correlation between $\mathbf{X} = \hat{\mathbf{K}}_1$ and $\mathbf{Y} = T_{\mathbf{u}}^{-1}(\hat{\mathbf{K}}_2)$.

Video identification has its own challenges that do not necessarily manifest for digital still images. The individual frames in a video are typically harshly quantized (compressed) to keep the video bit rate low. Thus, one needs many more frames for a good quality fingerprint estimate (e.g., thousands of frames). The compression artifacts carry over to the fingerprint estimate in a specific form of NUAs called "blockiness." Simple zero-meaning combined with Wiener filtering as described in Section 1.3.1.1 needs to be supplemented with a more aggressive procedure. The original paper [10] describes a notch filter that removes spikes due to JPEG compression in the frequency domain. The reader can also consult this source for an experimental verification of the fingerprint matching algorithm when applied to video clips.

1.4.4 Forgery detection

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 malicious 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 a blood stain.

The forgery detection algorithm tests for the presence of the fingerprint in each $B \times B$ sliding block $\mathcal{B}_b$ 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 a binary hypothesis testing problem:

$$\begin{aligned} H_0: &\quad \mathrm{W}_b = \boldsymbol{\Xi}_b, \\ H_1: &\quad \mathrm{W}_b = a_b \mathbf{I}_b \hat{\mathbf{K}}_b + \boldsymbol{\Xi}_b. \end{aligned} \tag{1.29}$$

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, a_b is an unknown attenuation factor due to possible processing of the forged image, and $\boldsymbol{\Xi}_b$ is the modeling noise assumed to be a white Gaussian noise with an unknown variance $\sigma^2_{\boldsymbol{\Xi},b}$. The likelihood ratio test for this problem is the normalized correlation

$$\rho_b = corr(\mathbf{I}_b \hat{\mathbf{K}}_b, \mathbf{W}_b). \tag{1.30}$$

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 ρ_b under both hypotheses. The probability density under H_0, $p(x|H_0)$, 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, $p(x|H_1)$, is much harder to obtain because it is heavily influenced by the block content. Dark blocks will have a lower value of the 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 a lower value of the 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 statistics ρ_b and its distribution would be if the block b was not tampered and indeed came from the camera.

The predictor is a mapping that needs to be constructed for each camera. The mapping assigns an estimate of the correlation $\hat{\rho}_b = \mathrm{Pred}(i_b, f_b, t_b)$ 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 $\mathrm{Pred}(.,.,.)$ can be constructed for example using regression [8, 11] or machine-learning techniques by training 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. For full-size camera images with several megapixels, blocks of 64×64 or 128×128 pixels seem to work well.

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

$$i_b = \frac{1}{|\mathcal{B}_b|} \sum_{i \in \mathcal{B}_b} \mathbf{I}(i). \tag{1.31}$$

We take as a measure of flatness the relative number of pixels, i, in the block whose sample intensity variance $\sigma^2_{\mathbf{I}}(i)$ estimated from the local 3×3 neighborhood of i is below a certain threshold

$$f_b = \frac{1}{|\mathcal{B}_b|} \Big| \{ i \in \mathcal{B}_b | \sigma_{\mathbf{I}}(i) < c\mathbf{I}(i) \} \Big|, \tag{1.32}$$

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

The texture measure evaluates the amount of edges in the block. Among many available options, we give the following example

$$t_b = \frac{1}{|\mathcal{B}_b|} \sum_{i \in \mathcal{B}_b} \frac{1}{1 + Var_5(\mathbf{H}(i))}, \tag{1.33}$$

where $Var_5(\mathbf{H}(i))$ is the sample variance computed from a local 5×5 neighborhood of pixel i for a high-pass filtered version of the block, $\mathbf{H} = H(\mathcal{B}_b)$, such as one obtained using an edge detector H.

Since one can obtain potentially hundreds of blocks from a single image, only a small number of images (e.g., ten) are needed to train (construct) the predictor. Fig. 1.6 shows the performance of the predictor for a Canon G2 camera. The form of the mapping Pred() was a second-order polynomial:

$$\text{Pred}(x, y, z) = \sum_{k+l+m \leq 2} \lambda_{klm} x^k y^l z^m, \tag{1.34}$$

where the unknown coefficients λ_{klm} were determined using a least squares estimator.

The data used for constructing the predictor can also be used to estimate the distribution of the prediction error v_b:

$$\rho_b = \hat{\rho}_b + v_b, \tag{1.35}$$

where $\hat{\rho}_b$ is the predicted value of the correlation for block b. Say that for a given block under investigation, we apply the predictor and obtain the estimated value $\hat{\rho}_b$. The distribution $p(x|H_1)$ is obtained by fitting a parametric probability density function to all points in Fig. 1.6 whose estimated correlation is in a small neighborhood of $\hat{\rho}_b$, $(\hat{\rho}_b - \epsilon, \hat{\rho}_b + \epsilon)$ for some small ϵ. A sufficiently flexible model that allows both thin and thick tails is the generalized Gaussian model with density $\alpha/(2\sigma\Gamma(1/\alpha)) \exp(-(|x - \mu|/\sigma)^\alpha)$ 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

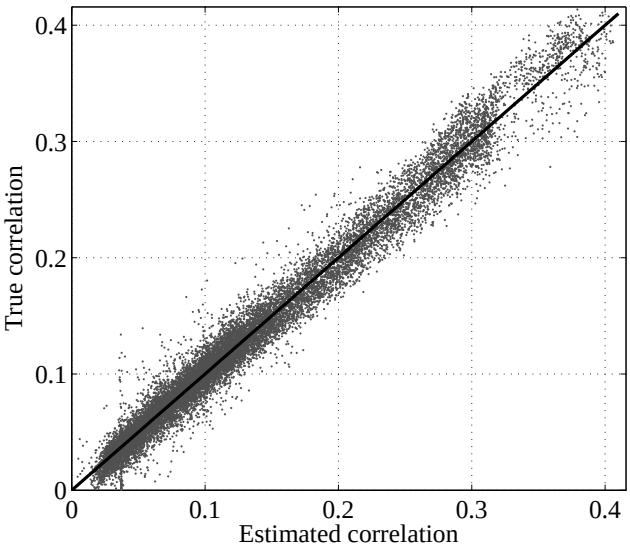

Fig. 1.6 Scatter plot of the true correlation ρ_b versus the estimate $\hat{\rho}_b$ for 30,000 128×128 blocks from 300 TIFF images from Canon G2.

the image and evaluates the test statistics ρ_b for each block b. The decision threshold τ for the test statistics ρ_b needs to be set to bound $\Pr(\rho_b > \tau | H_0)$ – the probability of misidentifying a tampered block as non-tampered. (In our experiments shown in Section 1.5.2, we requested this probability to be less than 0.01.)

Block b is marked as potentially tampered if $\rho_b < \tau$ 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 $\mathbf{Z}(i) = [\rho_b < \tau]$ ($[\cdot]$ is the Iverson bracket) indicating the potentially tampered pixels with $\mathbf{Z}(i) = 1$.

The above Neyman-Pearson criterion decides "tampered" whenever $\rho_b < \tau$ even though ρ_b may be "more compatible" with $p(x|H_1)$. This 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

$$p_{MD}(i) = \int_{-\infty}^{t} p(x|H_1)\mathrm{d}x. \tag{1.36}$$

Pixel i is labeled as non-tampered (we reset $\mathbf{Z}(i) = 0$) if $p_{MD}(i) > \beta$, where β is a user-defined threshold. (In experiments in the next section, $\beta = 0.01$.) The resulting binary map $\mathbf{Z}$ identifies the forged regions in their raw form.

The final map $\mathbf{Z}$ is obtained by post-processing $\mathbf{Z}$ using morphological filters. The block size imposes a lower bound on the size of tampered regions that the algorithm can identify. We thus 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 miss portions of the tampered boundary region.

1.5 Real-world examples

In this section, we demonstrate how the forensic methods proposed in the previous sections may be implemented in practice and also include some sample experimental results to give the reader an idea how the methods work on real imagery. The reader is referred to [9, 8, 11, 19] for more extensive tests and to [18] and [10] for experimental verification of device linking and fingerprint matching for video-clips. Camera identification from printed images appears in [20].

1.5.1 Camera identification

The experiment in this section was carried out on images from a Canon G2 camera with a four-megapixel CCD sensor. The camera fingerprint was estimated for each color channel separately using the maximum likelihood estimator (1.7) from 30 blue sky images acquired in the TIFF format. The estimated fingerprints were preprocessed as described in Section 1.3.1 to remove any residual patterns (NUAs) not unique to the sensor. This step is very important because these artifacts would cause unwanted interference at certain spatial shifts, $\mathbf{s}$, and scaling factors r, and thus 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 formula (1.12). All other images involved in this test were also converted to grayscale before applying the detectors described in Section 1.4.

The same camera was further used to acquire 720 images containing snapshots or various indoor and outdoor scenes under a wide range of light conditions and zoom settings spanning the 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 up to 50% in each dimen-

sion. 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 part was subsequently downsampled by a randomly chosen scaling ratio $r \in [0.5, 1]$. Finally, the images were converted to grayscale and compressed with 85% quality JPEG.

The detection threshold τ was chosen to obtain the probability of false alarm (1.28) $P_{FA} = 10^{-5}$. The camera identification algorithm was run with $r_{\min} = 0.5$ on all images. Only two missed detections were encountered (Fig. 1.7). 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.

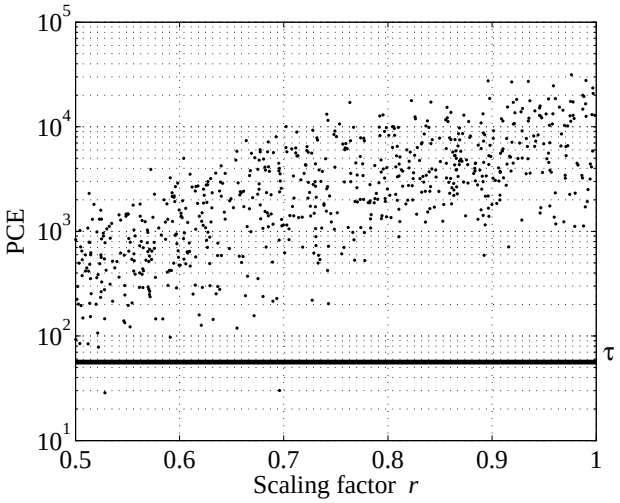

Fig. 1.7 PCE_{peak} as a function of the scaling ratio r for 720 images matching the camera. The detection threshold τ, which is outlined with a horizontal line, corresponds to $P_{FA} = 10^{-5}$.

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 before. 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 (Fig. 1.8).

The camera identification algorithm (without considering scaling and cropping) was subjected to a large scale test in [19]. The authors carried out experiments on 1,024,050 images from 6,896 cameras of 150 different models downloaded from the public image-sharing web site Flickr.com. With the

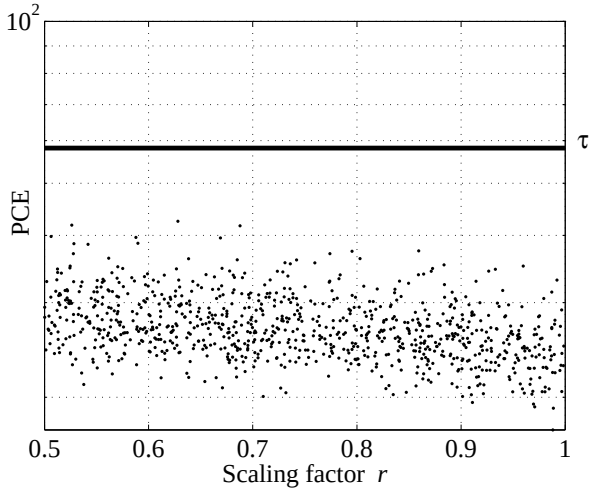

Fig. 1.8 PCE_{peak} for 915 images not matching the camera. The detection threshold τ is again outlined with a horizontal line and corresponds to $P_{FA} = 10^{-5}$.

decision threshold fixed to $\tau = 60$, the overall missed detection rate was $P_{MD} = 0.024$ with the false alarm rate $P_{FA} = 2.4 \times 10^{-5}$.

1.5.2 Forgery detection

The forgery-detection algorithm was tested on an image from a four-megapixel Olympus C765 digital camera equipped with a CCD sensor. Fig. 1.9 (a) shows the original image taken in the raw format. Using Photoshop, the girl in the middle was covered by pieces of the house siding from the background (b). The letters (c) – (f) show the result of the forgery-detection algorithm after the forged image was further processed using JPEG compression with quality factor 75 (c), after the forgery was subjected to denoising using a 3×3 Wiener filter with default value of σ in Matlab followed by JPEG compression with quality factor 90 (d), after processing using gamma correction with $\gamma = 0.5$ and again saved as JPEG 90 (e), and after downscaling to 60% of its size and JPEG 90 (f). In the last case, the image was upsampled back to its original size before the forgery-detection algorithm was applied (i.e., no search for the scaling ratio was performed). 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 automatically created forgeries as well as non-forged images, can be found in the original publications [9, 11].

Fig. 1.9 From upper left corner by rows: a) the original image, b) its tampered version, the result of the forgery-detection algorithm after the forged image was processed using c) JPEG with quality factor 75, d) 3×3 Wiener-filtered plus JPEG 90, e) gamma correction with $\gamma = 0.5$ plus JPEG 90, and f) scaled down by factor of 0.6 and JPEG 90.

1.6 Fighting the fingerprint-copy attack

Since the inception of camera identification methods based on sensor fingerprints in 2005 [41], researchers have realized that the sensor fingerprint can be copied onto an image that came from a different camera in an attempt to frame an innocent victim. In the most typical and quite plausible scenario, Alice, the victim, posts her images on the Internet. Eve, the attacker, estimates the fingerprint of Alice's camera and superimposes it onto another image. Indeed, as already shown in the original publication and in [17, 46], correlation detectors cannot distinguish between a genuine fingerprint and a fake one.

In this section, we describe a countermeasure [23, 12] against this fingerprint-copy attack that enables Alice to prove that she has been framed (that the fingerprint is fake). We assume that Alice owns a digital camera C. Eve takes

an image $\mathbf{J}$ from a different camera C' with fingerprint $\mathbf{K}' \neq \mathbf{K}$ and makes it appear as if it was taken by C. She does so by first estimating the fingerprint of C from some set of Alice's images and then properly adds it to $\mathbf{J}$.

In particular, let us assume that Eve has access to N images, from C and estimates its fingerprint $\hat{\mathbf{K}}_E$ using the estimator (1.7). We note that Eve certainly is free to use a different estimator. However, any estimation procedure will be some form of averaging of noise residuals $\mathbf{W}$. Having estimated the fingerprint, Eve may preprocess $\mathbf{J}$ to suppress the PRNU term $\mathbf{JK}'$ introduced by the sensor in C' and/or to remove any artifacts in $\mathbf{J}$ that are incompatible with C. Because suppressing the PRNU term is not an easy task [45], quite likely the best option for Eve is to skip this step altogether. This is because the PRNU component $\mathbf{JK}'$ in $\mathbf{J}$ is very weak to be detected per se and because it is unlikely that Alice will gain access to C'. In fact, Eve should avoid processing $\mathbf{J}$ too much as it may introduce artifacts of its own.

If Eve saves her forgery as a JPEG file, she needs to make sure that the quantization table is compatible with camera C, otherwise Alice will know that the image has been manipulated and did not come directly from her camera. If camera C' uses different quantization tables than C, Eve will inevitably introduce double-compression artifacts into $\mathbf{J}$, giving Alice again a starting point of her defense.

Unless C and C' are of the same model, the forged image may contain color-interpolation artifacts of C' incompatible with those of C. Alice could leverage techniques developed for camera brand/model identification [7] and prove that there is a mismatch between the camera model and the color interpolation artifacts. A knowledgeable attacker may, in turn, attempt to remove such artifacts of C' and introduce interpolation artifacts of C, for example, using the method described in [4].

It should now be apparent that it is far from easy to create a "perfect" forgery. While it is certainly possible for Alice to utilize traces of previous compression or color interpolation artifacts, no attempt is made in this section to exploit these discrepancies to reveal the forged fingerprint. Our goal is to develop techniques capable of identifying images forged by Eve even in the most difficult scenario for Alice when C' is of exactly the same model as C to avoid any incompatibility issues discussed above. Thus, Alice cannot take advantage of knowing any a priori information about C'.

The final step for Eve is to plant the estimated fingerprint in $\mathbf{J}$, creating thus the forged image $\mathbf{J}'$. In her attempt to mimic the acquisition process, and in accordance with (1.4), Eve superimposes the fake fingerprint multiplicatively, which is what would happen if $\mathbf{J}$ was indeed taken by C:

$$\mathbf{J}' = [\mathbf{J}(1 + \alpha\hat{\mathbf{K}}_E)], \tag{1.37}$$

where $\alpha > 0$ is a scalar fingerprint strength and $[x]$ is the operation of rounding x to integers forming the dynamic range of $\mathbf{J}$. Finally, Eve saves $\mathbf{J}'$ as

JPEG with the same or similar quantization table as that of the original image $\mathbf{J}$.

Formula (1.37) should be understood as three equations for each color channel of $\mathbf{J}$. This attack indeed succeeds in fooling the camera identification algorithm in the sense that the response of the fingerprint detector on $\mathbf{J}'$ (either the correlation (1.19), the generalized matched filter in [11], or the PCE (1.21)) will be high enough to indicate that $\mathbf{J}'$ was taken by camera C.

A very important issue for Eve is the choice of the strength α. Here, we grant Eve the ability to create a "perfect" forgery in the sense that $\mathbf{J}'$ elicits the same response of the fingerprint detector implemented with the true fingerprint $\mathbf{K}$ as when $\mathbf{J}'$ was indeed taken by C. A good estimate of the detector response can be obtained using a predictor, such as the one described in Section 1.4.4. This way, Eve makes sure that the fingerprint is not suspiciously weak or strong. While it is certainly true that Eve cannot easily construct the predictor because she does not have access to the true fingerprint $\mathbf{K}$, she may select the correct strength α by pure luck. To be more precise here, we grant Eve the ability to guess the right strength instead of giving her access to $\mathbf{K}$. We note that similar assumptions postulating a clairvoyant attacker are commonly made in many branches of information security.

1.6.1 Detecting fake fingerprints

Here, we describe a test using which Alice can decide whether an image came from her camera or whether it was forged by Eve as described above. For simplicity, we only discuss the case when Eve created one forged image $\mathbf{J}'$ and Alice has access to some of the N images used by Eve to estimate $\hat{\mathbf{K}}_E$ but Alice does not know which they are. She has a set of $N_c \geq N$ candidate images that Eve may have possibly used. This is a very plausible scenario because, unless Eve gains access to Alice's camera and takes images of her own and then removes them from the camera before returning the camera to Alice, Eve will have little choice but to use images taken by Alice, such as images posted by Alice on the Internet. In this case, Alice can prove that the forged image did not originally come from her camera by identifying among her candidate images those used by Eve.

We now explain the key observation based on which Alice can construct her defense. Let $\mathbf{I}$ be one of the N images available to Alice that Eve used to forge $\mathbf{J}'$. Since the noise residual of $\mathbf{I}$, $\mathbf{W_I}$, participates in the computation of $\hat{\mathbf{K}}_E$ through formula (1.7), $\mathbf{J}'$ will contain a scaled version of the *entire* noise residual $\mathbf{W_I} = \mathbf{IK} + \boldsymbol{\Xi}_\mathbf{I}$. Thus, besides the PRNU term, $\mathbf{W_I}$ and $\mathbf{W_{J'}}$ will share another signal – the noise $\boldsymbol{\Xi}_\mathbf{I}$. Consequently, the correlation $c_{\mathbf{I},\mathbf{J}'} = corr(\mathbf{W_I}, \mathbf{W_{J'}})$ will be larger than what it would be if the only common signal between $\mathbf{I}$ and $\mathbf{J}'$ was the PRNU component (which would be the case

if $\mathbf{J}'$ was not forged). As this increase may be quite small and the correlation itself may fluctuate significantly across images, the test that evaluates the statistical increase must be calibrated. We call this test the triangle test.

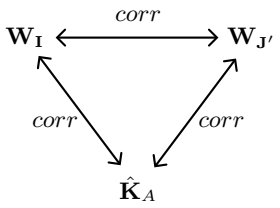

Fig. 1.10 Diagram for the triangle test.

Alice starts her defense by computing an estimate of the fingerprint of her camera $\hat{\mathbf{K}}_A$ from images guaranteed to not have been used by Eve. For instance, she can take new images with her camera C. Then, for a candidate image $\mathbf{I}$, she computes $c_{\mathbf{I},\mathbf{J}'}$, $c_{\mathbf{I},\hat{\mathbf{K}}_A} = corr(\mathbf{W_I}, \hat{\mathbf{K}}_A)$, and $c_{\mathbf{J}',\hat{\mathbf{K}}_A} = corr(\mathbf{W_{J'}}, \hat{\mathbf{K}}_A)$ (follow Fig. 1.10). The test is based on the fact that for images $\mathbf{I}$ that were not used to forge $\mathbf{J}'$, the value of $c_{\mathbf{I},\mathbf{J}'}$ can be estimated from $c_{\mathbf{I},\hat{\mathbf{K}}_A}$and $c_{\mathbf{J}',\hat{\mathbf{K}}_A}$ while when $\mathbf{I}$ was used in the forgery, the correlation $c_{\mathbf{I},\mathbf{J}'}$ will be higher than its estimate.

In order to obtain a more accurate relationship, similar to our approach to forgery detection in Section 1.4.4, we will work by blocks of pixels, denoting the signals constrained to block b with subscript b. We adopt the model (1.4) for the noise residuals and a similar model for Alice's fingerprint:

$$\mathbf{W}_{\mathbf{I},b} = a_{\mathbf{I},b}\mathbf{I}_b\mathbf{K}_b + \boldsymbol{\Xi}_{\mathbf{I},b}, \tag{1.38}$$

$$\mathbf{W}_{\mathbf{J}',b} = a_{\mathbf{J}',b}\mathbf{J}'_b\mathbf{K}_b + \boldsymbol{\Xi}_{\mathbf{J}',b}, \tag{1.39}$$

$$\hat{\mathbf{K}}_A = \mathbf{K}_b + \boldsymbol{\xi}_b. \tag{1.40}$$

The block-dependent attenuation factor a in (1.39) has been introduced due to the fact that various image processing that might have been applied by Eve to $\mathbf{J}$ may affect the PRNU factor differently in each block.

When $\mathbf{I}$ was not used by Eve, under some fairly mild assumptions about the noise terms in (1.39), the following estimate of $c_{\mathbf{I},\mathbf{J}'}$ is derived in [12, 23]

$$\hat{c}_{\mathbf{I},\mathbf{J}'} = corr(\mathbf{W_I}, \hat{\mathbf{K}}_A)corr(\mathbf{W_{J'}}, \hat{\mathbf{K}}_A)\mu(\mathbf{I}, \mathbf{J}')q^{-2}, \tag{1.41}$$

where $\mu(\mathbf{I}, \mathbf{J}')$ is the "mutual-content factor,"

$$\mu(\mathbf{I}, \mathbf{J}') = \frac{\sum_b a_{\mathbf{I},b}a_{\mathbf{J}',b}\overline{\mathbf{I}_b\mathbf{J}'_b}}{\sum_b a_{\mathbf{I},b}\overline{\mathbf{I}_b} \cdot \sum_b a_{\mathbf{J}',b}\overline{\mathbf{J}'_b}}N_B, \tag{1.42}$$

and the bar denotes the sample mean as before. The integer N_B is the number of blocks and $q \leq 1$ is the quality of $\hat{\mathbf{K}}_A$, $q^{-2} = 1 + (SNR_{\hat{\mathbf{K}}_A})^{-1}$, $SNR_{\hat{\mathbf{K}}_A} = \|\mathbf{K}\|^2 / \|\boldsymbol{\xi}\|^2$.

The attenuation factors can be estimated by computing the following block-wise correlations:[1]

$$a_{\mathbf{I},b} = \frac{\|\mathbf{W}_{\mathbf{I},b}\|}{\sqrt{\overline{\mathbf{I}_b^2}}\left\|\hat{\mathbf{K}}_{A,b}\right\|} corr(\mathbf{W}_{\mathbf{I},b}, \hat{\mathbf{K}}_{A,b}) q^{-2}. \tag{1.43}$$

Continuing the analysis of the case when $\mathbf{I}$ was not used by Eve, we consider $c_{\mathbf{I},\mathbf{J}'}$ and $\hat{c}_{\mathbf{I},\mathbf{J}'}$ as random variables over different images $\mathbf{I}$ for a fixed $\mathbf{J}'$. The dependence between these two random variables is well fit with a straight line $c_{\mathbf{I},\mathbf{J}'} = \lambda \hat{c}_{\mathbf{I},\mathbf{J}'} + c_0$. Because the distribution of the deviation from the linear fit does not seem to vary with $\hat{c}_{\mathbf{I},\mathbf{J}'}$ (see Fig. 1.11), we make a simplifying assumption that the conditional probability

$$\Pr(c_{\mathbf{I},\mathbf{J}'} - \lambda \hat{c}_{\mathbf{I},\mathbf{J}'} - c_0 = x | \hat{c}_{\mathbf{I},\mathbf{J}'}) \approx f_{\mathbf{J}'}(x), \tag{1.44}$$

is independent of $\hat{c}_{\mathbf{I},\mathbf{J}'}$.

When $\mathbf{I}$ was used by Eve in the multiplicative forgery, due to the additional common signal $\boldsymbol{\Xi}_{\mathbf{I}}$, the correlation $c_{\mathbf{I},\mathbf{J}'}$ increases to $\beta c_{\mathbf{I},\mathbf{J}'}$, where β is the following multiplicative factor derived in [23, 12]

$$\beta = 1 + \frac{\alpha}{N} \frac{\sum_b a_{\mathbf{J}',b} \overline{\mathbf{J}'_b} \|\boldsymbol{\Xi}_{\mathbf{I},b}\|^2}{\sum_b a_{\mathbf{I},b} a_{\mathbf{J}',b} \overline{\mathbf{I}_b \mathbf{J}'_b} \|\mathbf{K}_b\|^2}. \tag{1.45}$$

Notice that the percentual increase is proportional to the fingerprint strength α and the energy of the common noise component $\boldsymbol{\Xi}_{\mathbf{I},b}$; it is inversely proportional to N.

Alice now runs the following composite binary hypothesis test for every candidate image $\mathbf{I}$ from her set of N_c candidate images:

$$\begin{aligned} H_0 &: \quad c_{\mathbf{I},\mathbf{J}'} - \lambda \hat{c}_{\mathbf{I},\mathbf{J}'} - c_0 \sim f_{\mathbf{J}'}(x), \\ H_1 &: \quad c_{\mathbf{I},\mathbf{J}'} - \lambda \hat{c}_{\mathbf{I},\mathbf{J}'} - c_0 \nsim f_{\mathbf{J}'}(x). \end{aligned} \tag{1.46}$$

The reason why (1.46) cannot be turned into a simple hypothesis test is that the distribution of $c_{\mathbf{I},\mathbf{J}'}$ when $\mathbf{I}$ is used for forgery is not available to Alice and it cannot be determined experimentally because Alice does not know the exact actions of Eve. Thus, we resort to the Neyman-Pearson test and set our decision threshold τ to bound the probability of false alarm,

$$\Pr(c_{\mathbf{I},\mathbf{J}'} - \lambda \hat{c}_{\mathbf{I},\mathbf{J}'} - c_0 > \tau | H_0) = P_{FA}. \tag{1.47}$$

[1] Equation (1.43) holds independently of whether or not $\mathbf{I}$ was used by Eve.

The pdf $f_{\mathbf{J}'}(x)$ is often very close to a Gaussian but for some images $\mathbf{J}'$, the tails exhibit a hint of a polynomial dependence. Thus, to be conservative, we used Student's t-distribution for the fit.

Note that, depending on $\mathbf{J}'$, the constant of proportionality $\lambda > 1$, which suggests the presence of an unknown multiplicative hidden parameter in (1.41) most likely due to some non-periodic NUAs that were not removed using zero-meaning as described in Section 1.3.1. The quality of Alice's fingerprint, q, can be considered unknown (or simply set to 1) as different q will just correspond to a different λ (scaling of the x axis in the diagram of $c_{\mathbf{I},\mathbf{J}'}$ versus $\hat{c}_{\mathbf{I},\mathbf{J}'}$).

Alice now has two options. She can test each candidate image $\mathbf{I}$ separately by evaluating its p-value and thus, on a certain level of statistical significance, identify those images that were used by Eve for estimating her fingerprint. Alternatively, Alice can test for N_c candidate images $\mathbf{I}$ all at once whether $c_{\mathbf{I},\mathbf{J}'} - \lambda\hat{c}_{\mathbf{I},\mathbf{J}'} - c_0 \sim f_{\mathbf{J}'}(x)$. This "pooled test" will be a better choice for her for large N when the reliability of the triangle test for individual images becomes low.

1.6.2 Experiments

In this chapter, we report the results of experiments when testing individual images. The original publications [23, 12] contain much more detailed experimental evaluation including the pooled test and another case when multiple forged images are analyzed.

In the experiment, the signals entering the triangle test were preprocessed by zero-meaning. Wiener filtering, as described in Section (1.3.1) to suppress the NUAs, was only applied to $\mathbf{W}_{\mathbf{J}'}$ and not to $\mathbf{W}_{\mathbf{I}}$ to save computation time. The camera C' is the four-megapixel Canon PS A520 while C is Canon PS G2, which has the same native resolution. Both cameras were set to take images at the highest quality JPEG compression and the largest resolution. The picture-taking mode was set to "auto."

The fingerprint estimation algorithm and the forging algorithm depend on a large number of parameters, such as the parameters of the denoising filter. In presenting our test results, we intentionally opted for what we consider to be the most advantageous setting for Eve and the hardest one for Alice. This way, the results will be on the conservative side. Furthermore, to obtain a compact yet comprehensive report on the performance of the triangle test, the experiments were designed to show the effect of only the most influential parameters. To estimate her fingerprint $\hat{\mathbf{K}}_E$, Eve uses the most accurate estimator she can find in the literature (1.7) implemented using the denoising filter F described in [43] with the wavelet-domain Wiener filter parameter $\sigma = 3$ (valid for 8-bit per channel color images). From our experiments,

the reliability of the triangle test is insensitive to the denoising filter or the mismatch between the filters used by Eve and Alice.

Then, Eve forges a 24-bit color image $\mathbf{J}$ from camera C' to make it look as if it came from camera C. She first slightly denoises $\mathbf{J}$ using the same denoising filter F (with its Wiener filter parameter $\sigma = 1$) to suppress the fingerprint from camera C' and possibly other artifacts introduced by C'. The filter is applied to each color channel separately. Then, Eve adds the fingerprint to $\mathbf{J}$, using (1.37), and saves the result as JPEG with quality factor 90, which is slightly smaller than the typical qualities of the original JPEG images $\mathbf{J}$. The fingerprint strength factor α is determined so that the response of the generalized matched filter (Equation (11) in [11]) matches its prediction obtained using the predictor as described in Section (1.4.4). The predictor was implemented as a linear combination of intensity (1.31), texture (1.33), and flattening features (1.32), and their second-order terms. The coefficients of the linear fit were determined from 20 images of natural scenes using the least square fit. Note that because Eve needs to adjust α so that the JPEG-compressed $\mathbf{J}'$ elicits the same GMF value as the prediction, the proper value of α must be found, e.g., using a binary search. Finally, the true fingerprint $\mathbf{K}$ was estimated from 300 JPEG images of natural scenes.

On the defense side, Alice estimates her fingerprint $\hat{\mathbf{K}}_A$ from $N_A = 15$ blue-sky raw images (fingerprint quality was $q = 0.56$). Surprisingly, the quality of $\hat{\mathbf{K}}_A$ has little impact on the triangle test. Tests with $N_A = 70$ produced essentially identical results. In particular, it is not necessary for Alice to work with a better quality fingerprint than Eve! In all experiments, the block size was 128×128 pixels. The triangle test performed equally well for blocks as small as 64×64 and as large as 256×256.

We now evaluate the performance of the triangle test when applied to each candidate image individually. By far the most influential element is the number of images used by Eve, N, and the content of the forged image $\mathbf{J}$. Six randomly selected test images $\mathbf{J}$ shown in Fig. 1.12 were tested with the values of $N \in \{20, 50, 100, 200\}$. To give the reader a sense of the extent of Eve's forging activity, in Table 1.1 we report the PSNR between $\mathbf{J}$ and $\mathbf{J}'$ before it is JPEG compressed. The PSNR between $\mathbf{J}'$ and $F(\mathbf{J}')$ measures the total distortion that includes the slight denoising, $F(\mathbf{J})$, and quantization to 24-bit colors after adding the fingerprint. The PSNR between $\mathbf{J}'$ and the slightly denoised $F(\mathbf{J})$ measures the energy of the PRNU term only.

To accurately estimate the probability distribution $f_{\mathbf{J}}(x)$, we used 358 images from camera C that were for sure not used by Eve. All these images were taken within a period of about four years. In practice, depending on the situation, statistically significant conclusions may be obtained using a much smaller sample.

Fig. 1.11 presents a typical plot of $c_{\mathbf{I},\mathbf{J}'}$ versus $\hat{c}_{\mathbf{I},\mathbf{J}'}$ for $N = 20$ and $N = 100$. As expected, the separation between images used by Eve and those not used deteriorates with increasing N. When applying the triangle test individually to each candidate image, after setting the decision threshold

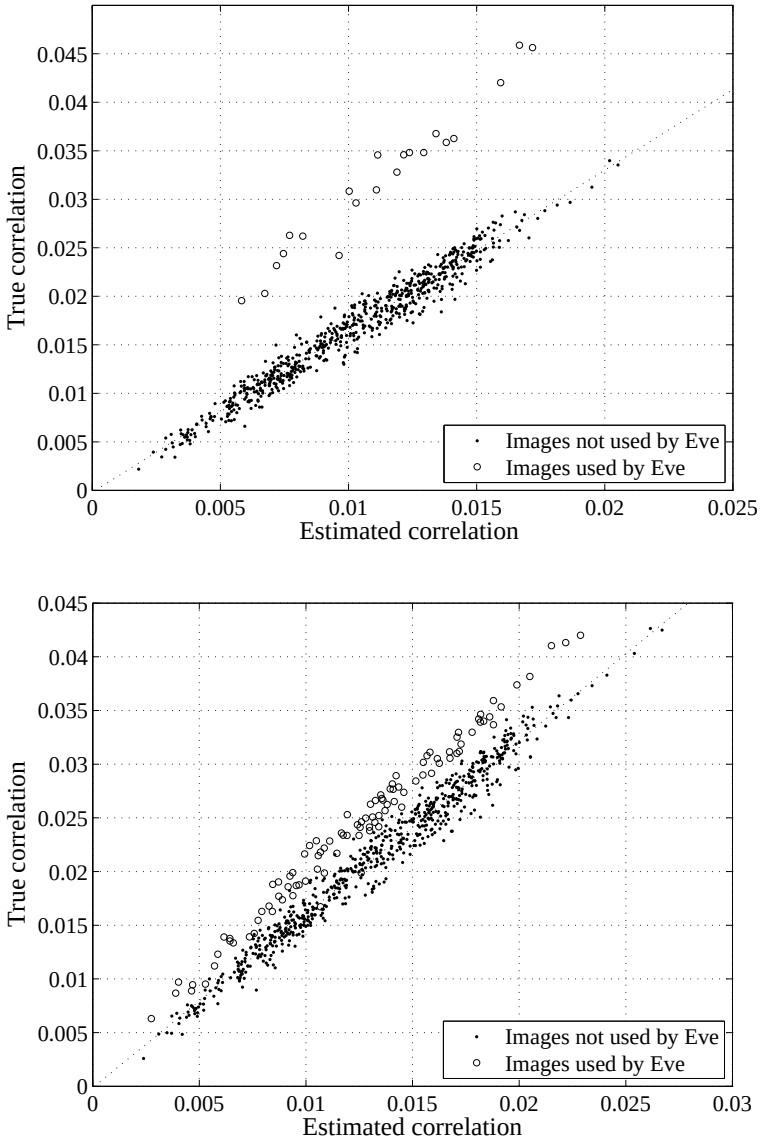

Fig. 1.11 True correlation $c_{\mathbf{I},\mathbf{J}'}$ versus the estimate $\hat{c}_{\mathbf{I},\mathbf{J}'}$ for image no. 5. Eve's fingerprint was estimated from $N = 20$ images (top) and $N = 100$ (bottom).

to satisfy a desired probability of false alarm, P_{FA}, the probability of correct detection P_D in the hypothesis test (1.46) is shown in Table 1.2. Each value of P_D was obtained by running the entire experiment as explained in Section 1.6.1 and evaluating the p-values for all images used by Eve.

#	$PSNR(F(\mathbf{J}),\mathbf{J}')$ [dB]				$PSNR(\mathbf{J},\mathbf{J}')$ [dB]			
N	20	50	100	200	20	50	100	200
1	48.8	51.8	53.2	53.9	47.6	49.5	50.3	50.7
2	49.0	51.8	53.1	53.8	47.8	49.8	50.6	50.9
3	50.1	51.8	52.9	53.4	48.7	49.8	50.5	50.8
4	54.5	56.4	57.5	58.7	49.5	50.0	50.2	50.4
5	49.5	52.2	53.2	54.0	47.7	49.3	49.7	50.1
6	50.8	53.2	54.3	55.1	49.3	51.0	51.6	52.1

Table 1.1 PSNR between the original image $\mathbf{J}$ and the forgery $\mathbf{J}'$ before JPEG compression for six test images.

#	P_D [%] for $P_{FA}=10^{-3}$				P_D [%] for $P_{FA}=10^{-4}$			
N	20	50	100	200	20	50	100	200
1	100	92	63	15	100	80	44	6
2	100	84	40	5	100	74	26	0
3	95	78	35	4	95	66	14	0
4	95	64	21	3	95	42	8	1
5	100	90	56	11	100	82	41	2
6	100	94	59	14	100	90	40	2

Table 1.2 Detection rate in percents for six text images.

The lower detection rate for image no. 4 is due to the low energy of the fingerprint (see the corresponding row in Table 1.1) dictated by the predictor. Because the image has smooth content, which is further smoothened by the denoising filter, the fingerprint PSNR in the noise residual $\mathbf{W}$ is higher than for other images. Consequently, a low fingerprint energy is sufficient in matching the predicted correlation. Image no. 3 also produced lower detection rates, mostly due to the fact that 26% of the image content is overexposed (the entire sky) with fully saturated pixels. The attenuation factor a_b in (1.39) is thus effectively equal to zero for such blocks b, while it is estimated in (1.43) under H_1 as being relatively large due to the absence of the noise term $\boldsymbol{\Xi}_{\mathbf{J}',b}$. A possible remedy is to apply the triangle test only to the non-saturated part of the image. However, then we experience a lower accuracy again due to a smaller number of pixels in the image. At this point, we note that if the attacker makes the forgery using (1.37) without attenuating the PRNU in saturated areas, the fingerprint will be too strong there, which could be used by Alice to argue that the fingerprint has been artificially added and the image did not come from her camera.

Fig. 1.12 Six original images **J** from a Canon PS A520 numbered by rows (#1, #2, #3); (#4, #5, #6).

We conclude this section with the statement that while it is possible to maliciously add a sensor fingerprint to an image to frame an innocent victim, adding it so that no traces of the forging process are detectable appears rather difficult. The adversary, Eve, will likely have to rely on images taken by Alice that she decided to share with others, for example on her Facebook site. However, the estimation error of the camera fingerprint estimated from such images will contain remnants of the entire noise residual from all images used by Eve. This fact is the basis of the triangle test using which Alice can identify the images that Eve used for her forgery and, in doing so, prove her innocence.

As a final remark, we note that while the reliability of the single-image triangle test breaks up at approximately $N = 100$, the test in its pooled form (which is described in [23, 12]) can be applied even when Eve uses a high-quality fingerprint estimated from more than 300 images. The reliability in general decreases with increasing ratio N_c/N.

The reliability of the triangle test can be somewhat decreased by modifying the fingerprint estimation process to limit outliers in the noise residuals of images used for the estimation [5, 39]. This will generally suppress the remnants of the noise residual used by the triangle test. More extensive tests, however, appear necessary to investigate the overall impact of this counter–counter-counter measure on the reliability of the identification algorithm when the fingerprint is not faked.

1.7 Temporal forensics

The goal of temporal forensics is to establish causal relationship among two or more objects [42, 44]. In this section, we show how pixel defects can be used to order images by their acquisition time given a set of images from the same camera whose time ordering is known.[2] Even though temporal data is typically found in the EXIF header, it may be lost when processing or editing images off-line. Additionally, since the header is easily modifiable, it cannot be trusted. Developing reliable methods for establishing temporal order between individual pieces of evidence is of interest for multiple reasons. Such techniques can help reveal deception attempts of an adversary or a criminal. The causal relationship also provides information about the whereabouts of the photographer.

Strong point defects (hot/stuck pixels) occur randomly in time and space on the sensor independently of each other [38, 36, 37, 13], which makes them useful for determining an approximate age of digital photographs. New defects appear suddenly and with a constant rate (It is a Poisson process.), which means that the time between the onsets of new defects follows the exponential distribution. Once a defect occurs, it becomes a permanent part of the sensor and does not heal itself.

The main cause of new pixel defects is environmental stress, primarily due to impacting cosmic rays. In general, smaller pixels are more vulnerable to point defects than larger pixels. Sensors age both at high altitudes and at the sea level. They do age faster at high altitudes or during airplane trips where cosmic radiation is stronger. Consequently, in real life the defect accumulation may not be linear in time.

The main technical problem for using point defects for temporal forensics is that such defects may not be easily detectable in individual images, depending on the image content, camera settings, exposure time, etc. In the next section, we describe a method for estimating pixel defect parameters and use it for determining an approximate age of a digital photograph using the principle of maximum likelihood.

1.7.1 Estimating point defects

Even though sensor defects can be easily estimated in a laboratory environment by taking test images under controlled conditions, a forensic analyst must work with a given set of images taken with camera settings that may be quite unfavorable for estimating certain defects. For example, the dark current is difficult to estimate reliably from images of bright scenes taken with a short exposure time and low ISO.

[2] Temporal ordering of digital images was for the first time considered in [42].

Let $\mathbf{Y}_k$, $k = 1, \ldots, d$, be d images of regular scenes taken at known times $t_1, \ldots, t_d$. As in the previous section, we work with noise residuals $\mathbf{W}_k = \mathbf{Y}_k - F(\mathbf{Y}_k)$ obtained using a denoising filter F. Since hot pixels (and stuck pixels with a large offset) are spiky in nature, denoising filters of the type [43] that extract additive white Gaussian noise are likely a poor choice for estimating point defects. Instead, non-linear filters, such as the median filter, are more likely to extract the spike correctly.

We use the model of pixel output (1.3):

$$\mathbf{W}_k(i) = \mathbf{I}_k(i)\mathbf{K}(i) + \tau_k \mathbf{D}(i) + \mathbf{c}(i) + \boldsymbol{\Xi}_k(i), \tag{1.48}$$

where k and i are the image and pixel indices, respectively. For simplicity, we model $\boldsymbol{\Xi}_k(i)$, $k = 1, \ldots, d$, as an i.i.d. Gaussian sequence with zero mean and variance $\sigma^2(i)$. The known quantities in the model are $\mathbf{I}_k \approx F(\mathbf{Y}_k)$ and τ_k; $\mathbf{W}_k$ are observables.

From now on, all derivations will be carried out for a fixed pixel i. This will allow us to drop the pixel index and make the expressions more compact. The unknown vector parameter $\boldsymbol{\theta} = (\mathbf{K}, \mathbf{D}, \mathbf{c}, \sigma)$ (for a fixed pixel, $\boldsymbol{\theta} \in \mathbb{R}^4$) can be estimated using the Maximum Likelihood (ML) principle:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} L(\mathbf{W}_1, \ldots, \mathbf{W}_d | \boldsymbol{\theta}), \tag{1.49}$$

where L is the likelihood function

$$L(\mathbf{W}_1, \ldots, \mathbf{W}_d | \boldsymbol{\theta}) = (2\pi\sigma^2)^{-d/2} \exp\left(-\frac{1}{2\sigma^2} \sum_{k=1}^{d} (\mathbf{W}_k - \mathbf{I}_k\mathbf{K} - \tau_k\mathbf{D} - \mathbf{c})^2 \right). \tag{1.50}$$

Because the modeling noise $\boldsymbol{\Xi}_k$ is Gaussian, the ML estimator becomes the least squares estimator [31]. Additionally, the model linearity allows us to find the maximum in (1.54) in the following well-known form:

$$(\hat{\mathbf{K}}, \hat{\mathbf{D}}, \hat{\mathbf{c}}) = (\mathbf{H}\mathbf{H}')^{-1}\mathbf{H}'\mathbf{W}, \tag{1.51}$$

$$\hat{\sigma}^2 = \frac{1}{d} \sum_{k=1}^{d} \left(\mathbf{W}_k - \mathbf{I}_k\hat{\mathbf{K}} - \tau_k\hat{\mathbf{D}} - \hat{\mathbf{c}} \right)^2, \tag{1.52}$$

where

$$\mathbf{H} = \begin{pmatrix} \mathbf{I}_1(i) & \tau_1 & 1 \\ \mathbf{I}_2(i) & \tau_2 & 1 \\ \cdots & \cdots & \cdots \\ \mathbf{I}_d(i) & \tau_d & 1 \end{pmatrix} \tag{1.53}$$

is a matrix of known quantities and $\mathbf{W} = (\mathbf{W}_1, \ldots, \mathbf{W}_d)'$ the vector of observations (noise residuals).

1.7.2 Determining defect onset

We now extend the estimator derived above to the case when we have observations (pixel intensities) across some time span during which the pixel becomes defective. Our goal is to estimate $\boldsymbol{\theta}$ before and after the onset, $\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}$, and the onset time j. The estimator derived in the previous section easily extends to this case

$$(\hat{\boldsymbol{\theta}}^{(0)}, \hat{\boldsymbol{\theta}}^{(1)}, \hat{j}) = \arg \max_{(\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, j)} L_j(\mathbf{W}_1, \ldots, \mathbf{W}_d | \boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}), \tag{1.54}$$

where

$$L_j(\mathbf{W}_1, \ldots, \mathbf{W}_d | \boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}) = L(\mathbf{W}_1, \ldots, \mathbf{W}_j | \boldsymbol{\theta}^{(0)}) \times L(\mathbf{W}_{j+1}, \ldots, \mathbf{W}_d | \boldsymbol{\theta}^{(1)}) \tag{1.55}$$

is the likelihood function written in terms of (1.50). Because of the form of (1.55), the maximization in (1.54) factorizes:

$$\begin{aligned}(\hat{\boldsymbol{\theta}}^{(0)}, \hat{\boldsymbol{\theta}}^{(1)}, \hat{j}) = \arg \max_j \Big\{ & \arg \max_{\boldsymbol{\theta}^{(0)}} L(\mathbf{W}_1, \ldots, \mathbf{W}_j | \boldsymbol{\theta}^{(0)}) \times \\ & \arg \max_{\boldsymbol{\theta}^{(1)}} L(\mathbf{W}_{j+1}, \ldots, \mathbf{W}_d | \boldsymbol{\theta}^{(1)}) \Big\},\end{aligned} \tag{1.56}$$

which converts the problem of onset estimation to the problem of estimating pixel defect addressed in the previous section.

1.7.3 Determining approximate acquisition time

We are now ready to develop the algorithm for placing a given image $\mathbf{I}$ under investigation among other d images, $\mathbf{I}_1, \ldots, \mathbf{I}_d$, whose time of acquisition is *known*, monotone increasing, and whose pixel defects are known, including the onset time and the parameters before and after the onset. This problem is again addressed using the ML approach. This time, only the time index j of $\mathbf{I}$ is the unknown as the parameters of all defective pixels are already known. Denoting the set of all defective pixels $\mathcal{D}$, the estimator becomes:

$$\hat{j} = \arg \max_j \prod_{i \in \mathcal{D}} L(\mathbf{W}_{\mathbf{I}}(i) | \boldsymbol{\theta}^{[j > j(i)]}(i)) \tag{1.57}$$

written in terms of (1.50). The superscript of $\boldsymbol{\theta}$ is the Iverson bracket.

1.7.4 Performance example

We applied the approach outlined above to images from a Canon PS sd400 digital camera. Total of $d = 329$ images with known acquisition times spanning almost 900 days were used to estimate the defect parameters for 46 point defects. Then, the acquisition times were estimated for 159 other images. The accuracy with which one can estimate the time obviously depends on how many new defects appear during the entire time span.

The noise residual $\mathbf{W}$ was extracted using a 3×3 median filter. Fig. 1.13 shows the true date versus the estimated date for all 159 images. The circles on the diagonal correspond to the training set of 329 images with known dates. They show the temporal distribution of training images. Note that no training images appear between time 400 and 500, limiting thus our ability to date images within this time interval.

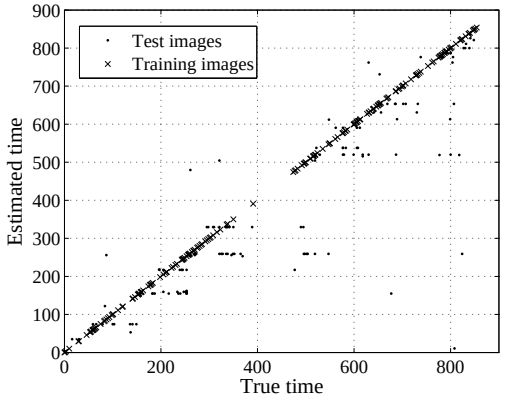

Fig. 1.13 True acquisition date versus date estimated using (1.57) based on detecting pixel defects. The average absolute error between the estimated and true date was 61.56 days.

1.7.5 Confidence intervals

In forensic setting, the analyst will likely be interested in statements of the type "the probability that image $\mathbf{I}$ was taken in time interval $[t, t']$ is at least p." The approach outlined above allows us to quantify the results in this way because the conditional probabilities $\Pr(\mathbf{W_I}|j) = \prod_{i \in \mathcal{D}} L(\mathbf{W_I}(i)|\boldsymbol{\theta}^{[j>j(i)]}(i))$ are known for each j. From the Bayes formula,

$$\Pr(j|\mathbf{W_I}) = \Pr(\mathbf{W_I}|j)\frac{\Pr(j)}{\Pr(\mathbf{W_I})}. \tag{1.58}$$

Thus, the probability that $\mathbf{I}$ was taken in time interval $[t, t']$ is

$$\Pr(j \in [t, t']|\mathbf{W_I}) = \frac{\sum_{k\in[t,t']} \Pr(\mathbf{W_I}|k)\Pr(k)}{\sum_k \Pr(\mathbf{W_I}|k)\Pr(k)}. \tag{1.59}$$

Depending on the situation at hand, the prior probabilities $\Pr(k)$ may be known from other forensic evidence or may be completely unknown. In the absence of any information about the priors, one could resort to the principle of maximum uncertainty and assume the least informative prior distribution – that the owner of the camera was taking images at a uniform rate, leading to $\Pr(k) = 1/(t_k - t_{k-1})$ for all k, where t_k is the time when the kth training image was taken.

1.8 Summary

Every imaging sensor contains defects and imperfections that can be utilized for a variety of forensic tasks. The photo-response non-uniformity (PRNU) plays the role of a fingerprint that well survives processing and can thus be used for identifying images taken by a camera whose fingerprint is known. Its absence in individual regions testifies about malicious processing applied to an image (forgery detection). The fingerprint can also serve as a template to recover previously applied geometrical processing. The methodology applies to CCD as well as CMOS sensors and to digital still and video cameras as well as scanners [25, 16, 34].

Although it is possible for an adversary to superimpose a camera fingerprint onto an image from a different camera and thus frame an innocent victim (the fingerpint-copy attack), it is not easy to do this without leaving detectable traces. The so-called triangle test can reveal when a fingerprint has been maliciously added and one can even identify the images that the adversary utilized in the attack.

Besides PRNU, sensors also contain the so-called point defects, examples of which are hot and stuck pixels and pixels with abnormal sensitivity. Such defects occur randomly on the sensor and randomly in time. This makes them useful for determining an approximate time when an image was taken. This chapter outlines a maximum-likelihood estimator of time of acquisition that basically detects the presence of point defects with a known onset in an image.

The performance of all forensic methods introduced here is briefly demonstrated on real images. Throughout the text, references to previously pub-

lished articles guide the interested reader to more detailed technical information.

1.8.1 Related publications

When the camera identification technique is deployed on a large scale, one quickly runs into complexity issues. For example, the seemingly routine task of matching an image (or fingerprint) to a large database of fingerprints stored in a database may be quite time consuming already when the database holds merely hundreds of fingerprints. This is because correlating each database fingerprint with the query signal may take hours even in the simplest case when no search for geometrical transformation is carried out. To address this problem, researchers [24] introduced the concept of a fingerprint digest and sparse data structures to cut down the searching time significantly.

A similar task of matching a large database of images to a small number of fingerprints was proposed in [2]. Here, the complexity was decreased using a hierarchical binary search.

For completeness, we note that there exist approaches combining sensor noise defects with machine-learning classification [25, 33, 6]. An older version of this forensic method was tested for cell phone cameras in [6] and in [47] where the authors show that a combination of sensor-based forensic methods with methods that identify camera brand can decrease false alarms. The improvement reported in [47], however, is unlikely to hold for the newer version of the sensor noise forensic method presented in this chapter as the results of [47] appear to be influenced by uncorrected non-unique artifacts discussed in Section 1.3.1. The problem of pairing of a large number of images was studied in [3] using an ad hoc approach. A large-scale experimental evaluation of camera identification on over one million images from over 6,800 cameras covering 150 models appears in [19]. Identification of printed images is the subject of [20]. Anisotropy of image noise for classification of images into scans, digital camera images, and computer art appeared in [33]. The effect of denoising filters on the performance of sensor-based camera identification was studied in [1].

1.8.2 Real-life applications

Camera identification based on sensor fingerprints described in this chapter passed the Daubert challenge `http://en.wikipedia.org/wiki/Daubert_standard` in the State of Alabama in July 2011. In March 2009, Miroslav Goljan testified in Scotland as an expert witness in a high-profile case that involved child abuse crimes by a pedophile ring. See the article “Operation Alge-

bra" at http://p10.hostingprod.com/@spyblog.org.uk/blog/2009/05/. Cases involving movie piracy and illegal acquisition and distribution of movies taped inside a movie theater form another area of possible applications of this technology. The sensor fingerprint can also be used to reverse-engineer in-camera geometrical processing, such as digital zoom or correction of lens distortion, and to provide a blind estimate of the focal length at which an image was taken [22].

ACKNOWLEDGMENT

The work on this chapter was partially supported by the NSF grant number CNF-0830528. 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 Office of Scientific Research or the U.S. Government. The author would like to thank Miroslav Goljan for useful discussions.

References

1. I. Amerini, R. Caldelli, V. Cappellini, F. Picchioni, and A. Piva. Analysis of denoising filters for photo-response non-uniformity noise extraction in source camera identification. In *Proc. of the 16th international conference on Digital Signal Processing*, pages 511–517, 2009.
2. S. Bayram, H. T. Sencar, and N. Memon. Efficient techniques for sensor fingerprint matching in large image and video databases. In N. Memon and J. Dittmann, editors, *Proceeedings of SPIE Electronic media Forensics and Security XII*, volume 7541, pages 09–01–09–12, January 2010.
3. G. J. Bloy. Blind camera fingerprinting and image clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):532–534, March 2008.
4. R. Böhme and M. Kirchner. Synthesis of color filter array pattern in digital images. In N. Memon and J. Dittmann, editors, *Proc. SPIE, Media Forensics and Security XI*, volume 7254, pages 0K–0L, San Jose, CA, January 18–22 2009.
5. R. Caldelli, I. Amerini, and A. Novi. An analysis on attacker actions in fingerprint-copy attack in source camera identification. In *Proc. IEEE WIFS*, Iguazu Falls, Brazil, November 29 – December 2 2011.
6. O. Çeliktutan, I. Avcibas, and B. Sankur. Blind identification of cellular phone cameras. In E.J. Delp and P.W. Wong, editors, *Proc. SPIE Electronic Imaging, Steganography, Security, and Watermarking of Multimedia Contents IX*, volume 6505, pages H1–H12, January 2007.
7. O. B. Celiktutan, B. Sankur, and I. Avcibas. Blind identification of source cell-phone model. *IEEE Transactions on Information Forensics and Security*, 3(3):553–566, 2008.
8. M. Chen, J. Fridrich, and M. Goljan. Digital imaging sensor identification (further study). In E.J. Delp and P.W. Wong, editors, *Proc. SPIE Electronic Imag-*

ing, Steganography, Security, and Watermarking of Multimedia Contents IX, volume 6505, pages 0P–0Q, January 2007.
9. M. Chen, J. Fridrich, and M. Goljan. Imaging sensor noise as digital x-ray for revealing forgeries. In T. Furon et al., editor, *Proc. 9th Information Hiding Workshop, Saint Malo, France*, volume 4567 of *LNCS*, pages 342–358. Springer-Verlag, June 2007.
10. M. Chen, J. Fridrich, and M. Goljan. Source digital camcorder identification using ccd photo response nonuniformity. In E.J. Delp and P.W. Wong, editors, *Proc. SPIE Electronic Imaging, Steganography, Security, and Watermarking of Multimedia Contents IX*, volume 6505, pages 1G–1H, January 2007.
11. M. Chen, J. Fridrich, and M. Goljan. Determining image origin and integrity using sensor noise. *IEEE Transactions on Information Forensics and Security*, 1(1):74–90, March 2008.
12. M. Chen, J. Fridrich, and M. Goljan. Defending against fingerprint-copy attack in sensor-based camera identification. *IEEE Transactions on Information Security and Forensics*, 2010. submitted.
13. J. Dudas, L. M. Wu, C. Jung, G. H. Chapman, Z. Koren, and I. Koren. In R. A. Martin, J. M. DiCarlo, and N. Sampat, editors, *Identification of in-field defect development in digital image sensors*, volume 6502, page 65020Y. SPIE, 2007.
14. T. Filler, J. Fridrich, and M. Goljan. Using sensor pattern noise for camera model identification. In *Proc. IEEE International Conference on Image Processing (ICIP)*, October 2008.
15. 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*, volume 3301-20, pages 168–177, January 1998.
16. T. Gloe, E. Franz, and A. Winkler. Forensics for flatbed scanners. In E.J. Delp and P.W. Wong, editors, *Proc. SPIE Electronic Imaging, Steganography, Security, and Watermarking of Multimedia Contents IX*, volume 6505, pages 1I–1J, January 2007.
17. T. Gloe, M. Kirchner, A. Winkler, and R. Boehme. Can we trust digital image forensics? In *Proc. The 15th International conference on Multimedia, Multimedia '07, ACM*, pages 78–86, 2007.
18. M. Goljan, M. Chen, and J. Fridrich. Identifying common source digital camera from image pairs. In *Proc. IEEE International Conference on Image Processing (ICIP)*, September 2007.
19. M. Goljan, T. Filler, and J. Fridrich. Camera identification – large scale test. In N. Memon and J. Dittmann, editors, *Proceeedings of SPIE Electronic media Forensics and Security XI*, volume 7254, pages 0I–01–0I–12, January 2009.
20. M. Goljan and J. Fridrich. Camera identification from printed images. In E.J. Delp, P. W. Wong, N. Memon, and J. Dittmann, editors, *Proceeedings of SPIE Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, volume 6819, pages OI1–OI12, January 2008.
21. M. Goljan and J. Fridrich. Camera identification from scaled and cropped images. In E. Delp, P. W. Wong, N. Memon, and J. Dittmann, editors, *Proceeedings of SPIE Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, volume 6819, pages OE1–OE13, January 2008.
22. M. Goljan and J. Fridrich. Sensor-fingerprint based identification of images corrected for lens distortion. In N. D. Memon, A. Alattar, and E. J. Delp, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security and Forensics 2012*, January 22–26, 2012.
23. M. Goljan, J. Fridrich, and M. Chen. Sensor noise camera identification: Countering counter-forensics. In N. Memon, J. Dittmann, and A. Alattar, editors, *Proc. SPIE Electronic Imaging, Steganography, Security, and Watermarking of Multimedia Contents XII*, volume 7541, pages 0S–01–0S–12, January 17–21 2010.

24. M. Goljan, J. Fridrich, and T. Filler. Managing a large database of camera fingerprints. In N. Memon, J. Dittmann, and A. Alattar, editors, *Proceeedings of SPIE Electronic media Forensics and Security XII*, volume 7541, pages 08–01–08–12, January 2010.
25. H. Gou, A. Swaminathan, and M. Wu. Robust scanner identification based on noise features. In E.J. Delp and P.W. Wong, editors, *Proc. SPIE Electronic Imaging, Steganography, Security, and Watermarking of Multimedia Contents IX*, volume 6505, pages 0S–0T, January 2007.
26. G. Healey and R. Kondepudy. Radiometric CCD camera calibration and noise estimation.
27. G. E. Healey and R. Kondepudy. Radiometric CCD camera calibration and noise estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):267–276, March 1994.
28. G. C. Holst. *CCD Arrays, Cameras, and Displays*. JCD Publishing & SPIE Press, USA, 2nd edition, 1998.
29. C. R. Holt. Two-channel detectors for arbitrary linear channel distortion. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-35(3):267–273, March 1987.
30. J.R. Janesick. *Scientific Charge-Coupled Devices*, volume PM83. SPIE Press Monograph, 2001.
31. S. M. Kay. *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*, volume II. Upper Saddle River, NJ: Prentice Hall, 1998.
32. S.M. Kay. *Fundamentals of Statistical Signal Processing, Detection Theory*, volume II. Prentice Hall, 1998.
33. N. Khanna, A. K. Mikkilineni, G. T. C. Chiu, J. P. Allebach, and E. J. Delp. Forensic classification of imaging sensor types. In E.J. Delp and P.W. Wong, editors, *Proc. SPIE Electronic Imaging, Steganography, Security, and Watermarking of Multimedia Contents IX*, volume 6505, pages U1–U9, January 2007.
34. N. Khanna, A.K. Mikkilineni, G.T.C. Chiu, J.P. Allebach, and E.J. Delp. Scanner identification using sensor pattern noise. In E.J. Delp and P.W. Wong, editors, *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, January.
35. K. Kurosawa, K. Kuroki, and N. Saitoh. CCD fingerprint method – identification of a video camera from videotaped images. In *Proc. IEEE International Conference on Image Processing (ICIP)*, pages 537–540, October 1999.
36. J. Leung, G. H. Chapman, I. Koren, and Z. Koren. Automatic detection of in-field defect growth in image sensors. In *DFT '08: Proceedings of the 2008 IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems*, pages 305–313, Washington, DC, 2008. IEEE Computer Society.
37. J. Leung, G. H. Chapman, I. Koren, and Z. Koren. Characterization of gain enhanced in-field defects in digital imagers. *Defect and Fault-Tolerance in VLSI Systems, IEEE International Symposium on*, 0:155–163, 2009.
38. J. Leung, J. Dudas, G. H. Chapman, I. Koren, and Z. Koren. Quantitative analysis of in-field defects in image sensor arrays. In *Defect and Fault-Tolerance in VLSI Systems, 2007. DFT '07. 22nd IEEE International Symposium on*, pages 526–534, September 2007.
39. C.-T. Li. Source camera identification using enhanced sensor pattern noise. In *Proc. of IEEE ICIP*, pages 7–11, 2009.
40. J. Lukáš, J. Fridrich, and M. Goljan. Determining digital image origin using sensor imperfections. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII*, pages 249–260, San Jose, CA, January 16–20, 2005.
41. J. Lukáš, J. Fridrich, and M. Goljan. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2):205–214, June 2006.

42. J. Mao, O. Bulan, G. Sharma, and S. Datta. Device temporal forensics: An information theoretic approach. In *Proc. IEEE International Conference on Image Processing (ICIP)*, volume 1, November 2009.
43. M. K. Mihcak, I. Kozintsev, K. Ramchandran, and P. Moulin. Low-complexity image denoising based on statistical modeling of wavelet coefficients. *IEEE Signal Processing Letters*, 6(12):300–303, December 1999.
44. A. D. Rosa, F. Uccheddu, A. Costanzo, A. Piva, and M. Barni. Exploring image dependencies: a new challange in image forensics. In N. Memon, J. Dittmann, and A. Alattar, editors, *Proceeedings of SPIE Electronic media Forensics and Security XII*, volume 7541, pages 0X–1–0X–12, January 2010.
45. K. Rosenfeld and H. T. Sencar. A study of the robustness of PRNU-based camera identification. In N. Memon, J. Dittmann, and A. Alattar, editors, *Proc. SPIE Electronic Imaging, Steganography, Security, and Watermarking of Multimedia Contents XI*, volume 7254, pages 0M–0N, January 18–22 2009.
46. M. Steinebach, H. Liu, P. Fan, and S. Katzenbeisser. Cell phone camera ballistics: attacks and countermeasures. In *Proc. SPIE, Multimedia on Mobile Devices 2010*, volume 7542, pages 0B–0C, San Jose, CA, January 18–22 2010.
47. Y. Sutcu, S. Bayram, H.T. Sencar, and N. Memon. Improvements on sensor noise based source camera identification. In *IEEE International Conference on Multimedia and Expo*, pages 24–27, July 2007.