Chapter 9

Image provenance inference through content-based device fingerprint analysis

Xufeng Lin and Chang-Tsun Li

9.1 Introduction

The last few decades have witnessed the increasing popularity of low-cost and high-quality digital imaging devices ranging from digital cameras to cellphones with built-in cameras, which makes the acquisition of digital images become easier than ever before. Meanwhile, the ever-increasing convenience of image acquisition has bred the pervasiveness of powerful image editing tools, which allow even unskilled persons to easily manipulate digital images. As a consequence, the credibility of digital images has been questioned and challenged. Under the circumstance where digital images serve as the critical evidence, e.g., presented as evidence in courts, being able to infer the provenance of an image becomes essential for recovering truth and ensuring justice.

As an important branch of digital forensics, image provenance inference aims to determine the original source of a digital image. The provenance of an image provides forensic investigators with rich information about the originality and integrity of the image. It does not only look for answers to the question of which device has been used to acquire a given image, but also conveys other implications of the credibility of an image. For example, the inconsistent provenance information from different regions of an image indicates that the image may have been tampered with. This chapter mainly introduces and discusses several intrinsic *device fingerprints* and their applications in image provenance inference. These fingerprints arise from either the hardware or software processing components in the image acquisition pipeline and exhibit themselves as specific patterns or traces in the image. Analyses of these fingerprints provide useful information for inferring the image provenance and uncovering underlying facts about the image.

In the remainder of this chapter, we will first discuss why the techniques that based on digital watermark and metadata are impractical or unreliable for image provenance inference in Section 9.2 and 9.3, respectively. In Section 9.4, we will introduce several intrinsic device fingerprints arising from different processing components in the imaging pipeline of a device. In Section 9.5, we will concentrate on Sensor Pattern Noise (SPN) and discuss in detail its applications in image prove-

nance inference. Section 9.6 concludes this chapter and points out several directions of future research.

9.2 Why Not Digital Watermark?

Digital watermark is an extra message that is embedded, usually in an invisible way, to digital contents like images, audio and video, for the purpose of protecting the ownership of digital contents. It offers an imperceptible way to insert digital object identifier, serial number or other image source information in host images, and thus provides a promising approach for inferring the provenance of an image. The *robust* watermark, which is able to survive a variety of image processing operations such as image compression, image filtering, and geometric modifications, can be used to verify the provenance of an image that has been redistributed over untrusted networks. The *fragile* or *semi-fragile* watermark [1–9], which is readily altered or destroyed when the host image is modified, has been intensively used to determine whether the image has been altered since its original recording.

In spite of the effectiveness of the techniques based on digital watermark, they can only be applied when the image is protected at the origin. Nowadays, the majority of images do not contain a digital watermark mainly due to the following reasons:

- Camera manufacturers have to devise extra digital watermark embedding components in the camera, so only some high-end cameras have watermark embedding features.

- The embedded watermark may degrade the image quality and significantly reduce the market value of cameras equipped with a watermark embedding component.

- The successful implementation of watermark-based protection requires close collaborations among publishers/manufaturers, investigators and potentially trusted third-party organizations. This restricts the wide adoption of digital watermark in digital devices.

9.3 Why Not Metadata?

Another way to determine the provenance of an image is through the use of metadata created by the source device. In particular, the Exchangeable Image File Format (EXIF) is the most ubiquitous metadata standard supported by many digital camera manufacturers. Part of the information retrieved from the EXIF header of an image is shown in Fig. 9.1. By accessing the EXIF header, some information, such as the "Make" and "Model" of the camera that has been used to take the image, can be retrieved. Other information, such as the "Create Date" and "Modify Date", can serve as useful clues to determine whether the image has been modified since its original recording. A few attempts [10, 11] have been made to exploit the EXIF information for forensic purposes.

Make	CASIO COMPUTER CO.,LTD.
Model	EX-Z150
Orientation	Horizontal (normal)
XResolution	72
YResolution	72
ResolutionUnit	inches
Software	1.00
ModifyDate	2009:01:06 12:39:41
YCbCrPositioning	Co-sited
	---- **ExifIFD** ----
ExposureTime	1/80
FNumber	10.0
ExposureProgram	Program AE
ISO	64
ExifVersion	0221
DateTimeOriginal	2009:01:06 12:39:41
CreateDate	2009:01:06 12:39:41
ComponentsConfiguration	Y, Cb, Cr, -
CompressedBitsPerPixel	4.098876437
ApertureValue	10.0

Figure 9.1: Part of the EXIF information retrieved from an image with an EXIF editing tool [12].

However, EXIF metadata is not always available. On the one hand, not all imaging devices and image file formats support EXIF standard, especially the older versions of devices and image formats. As a result, images taken with such devices or stored in such formats may not contain EXIF information. Early versions of image editing software, such as Adobe Photoshop 5.0, do not recognize the EXIF standard and strip the EXIF metadata when they resave the images. On the other hand, with the rise of photo sharing on social networks like Facebook, Instagram, and Twitter, there has been increasing concern and fear about the personal information embedded in the photos shared online. At the same time, the latest generation of cameras and phones are able to add location information or GPS coordinates to the EXIF metadata, which makes photo sharing a privacy hazard. For this reason, the EXIF metadata is stripped out by almost all the major social networks when the images are being uploaded.

Moreover, the EXIF metadata is easily removable or replaceable. Anyone who wishes to remove or edit EXIF metadata will find a range of tools at their disposal on the Internet. With these EXIF editing tools at hand, experienced photographers can even develop their own techniques to edit EXIF metadata. Therefore, even if EXIF metadata is present, it is not a reliable or trustworthy indicator of the image source.

9.4 Device Fingerprints

Illustrated in Fig. 9.2 is a simplified image acquisition pipeline in typical cameras. The light from the scene goes through the lens (usually covered by an associated color Filter array (CFA)) and projects on the surface of the sensor, which converts the optical signal into the raw image signal. Sequentially undergoing different processing components such as CFA interpolation, white balancing, camera response function (CRF), JPEG compression, etc. the raw image signal finally forms the image data suitable for visualization or display purpose. The final image carries spe-

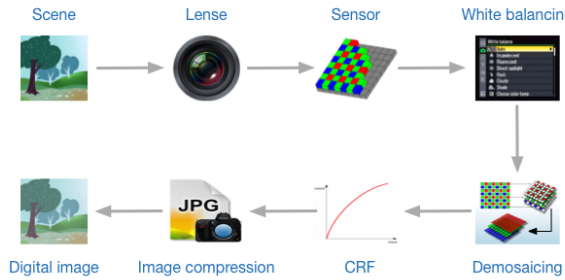


Figure 9.2: A simplified image acquisition pipeline in typical cameras.

cific patterns or traces left by each processing component in the image acquisition pipeline. Such patterns or traces are intrinsic to the imaging pipeline, so they can be considered as some sort of fingerprints of the source device and used for image provenance inference. It is similar to bullet scratches allow forensic investigators to match a bullet to a particular barrel. In the following subsections, we will introduce different device fingerprints arising from different processing components in the image acquisition pipeline.

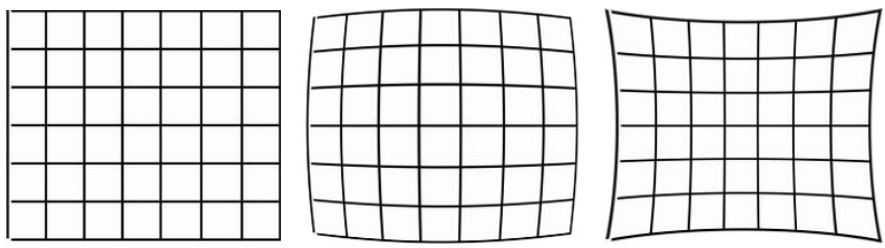
9.4.1 Optical Aberrations

Each camera is equipped with a complex optical system. In an ideal imaging system, the light rays from a point of an object pass through the lens and converge to a corresponding point on the sensor. However, realistic optical systems deviate from such an ideal model and introduce optical aberrations in the captured images. It should be noted that optical aberrations are caused by the optical specifications designed by device manufacturers (due to the wave nature of light) rather than any flaws in the optical elements. Different camera models are typically equipped with different optical systems, which have their own aberration characteristics. Therefore, the optical aberrations appear in an image can be used for inferring the provenance of the image or even verifying the content of the image. Optical aberrations can be categorized into different types, such as chromatic aberrations, spherical aberrations, coma and radial lens distortion. We refer readers to the Chapter 3 of [13] for a detailed description of each type of aberrations. Some of the optical aberrations have been exploited for image provenance inference. Johnson and Farid [14] modeled the lateral chromatic aberration , which occurs when different wavelengths of light do not converge to the same point on the sensor, as the expansion/contraction of the coordinates of

the red and blue channel with respect to the coordinate of the green channel:

$$\begin{pmatrix} x_r \\ y_r \end{pmatrix} = \begin{pmatrix} x_g - x_1 \\ y_g - y_1 \end{pmatrix} \alpha_1 + \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \tag{9.1}$$
$$\begin{pmatrix} x_b \\ y_b \end{pmatrix} = \begin{pmatrix} x_g - x_2 \\ y_g - y_2 \end{pmatrix} \alpha_2 + \begin{pmatrix} x_2 \\ y_2 \end{pmatrix},$$

where $(x_c, y_c), c \in \{r, g, b\}$ is the coordinate of channel c, and $(x_i, y_i), i \in \{1, 2\}$ and $\alpha_i, i \in \{1, 2\}$ are the coordinate of the center and the magnitude of distortion, respectively. In such a way, the red to green channel and blue to green channel distortions are characterized by the parameters (x_1, y_1, α_1) and (x_2, y_2, α_2), respectively. By maximizing the mutual information between each pair of color channels (i.e., the red and green channel, or the blue and green channel), the parameters can be globally estimated from the entire image. Any inconsistency between the globally estimated parameters and the parameters estimated locally from a suspect image region can be used as evidence of image forgery [14]. In [15], these six parameters estimated from images captured by four cellphones (two of them are of the same model) are used as features to train a support vector machine (SVM) classifier, which will be used for classifying images of unknown provenance. Testing on 180 images taken by three cellphones of different models shows an average classification accuracy of 92.2%. But the accuracy of differentiating the two cameras of the same model is as low as 50%, which is akin to a random guess.



(a) (b) (c)

Figure 9.3: Radial lens distortion of a rectangular grid. (a) Undistorted grid. (b) Barrel distortion. (c) Pincushion distortion.

Another pronounced and visually distinct aberration is the radial lens distortion, which arises from the fact that the magnification of an image is non-uniformly across the image plane but depends on the radial distance, r, from the optical center. When the magnification increases with r, the distortion is known as the "barrel distortion" (Fig. 9.3(b)). Conversely, it is known as the "pincushion distortion" (Fig. 9.3(c)), if the magnification decreases with r. Choi *et al.* [16, 17] adopted a simple polynomial model [18] to formulate the relationship between the distorted image coordinate (x_d, y_d) and undistorted image coordinate (x_u, y_u):

$$r_u = r_d + k_1 r_d^3 + k_2 r_d^5, \tag{9.2}$$

where $r_d = \sqrt{(x_d - x_0)^2 + (y_d - y_0)^2}$ and $r_u = \sqrt{(x_u - x_0)^2 + (y_u - y_0)^2}$ are the radius from the optical center (x_0, y_0) in the distorted and undistorted image, respectively. The parameters (k_1, k_2) characterizing the radial distortions can be estimated using the algorithm proposed in [18]. Similar to the work in [15], Choi *et al.* trained an SVM classifier using the distortion parameters estimated from images of different cameras and classified images of unknown provenance. On a small dataset consisting of 180 images taken by three cameras of different models, they reported an average accuracy of 91.5% [16]. The accuracy decreases to 89.1% on a larger dataset consisting of images from five cameras [17].

The results on small datasets show that the optical aberrations are promising in image provenance inference, but their limitations are apparent:

- Cameras of the same model are equipped with the same optical specification, therefore the optical aberrations are insufficient for identifying individual source cameras of the same model [15].

- The optical aberrations are closely related with camera settings, such as focal length [16, 17, 19], focal distance and even aperture size [20]. Different camera settings may introduce considerable intra-model variations and make the classification over a range of camera settings more problematic.

- The estimation of aberration parameters is influenced by JPEG compression, random noise and image cropping. Van *et al.* [15] showed that the classification accuracy declines when testing on images processed by some common image operations.

The above limitations of optical aberrations restrict their applicability to more diverse datasets in the sense of camera models, camera settings and image processing operations.

9.4.2 CFA and Demosaicing

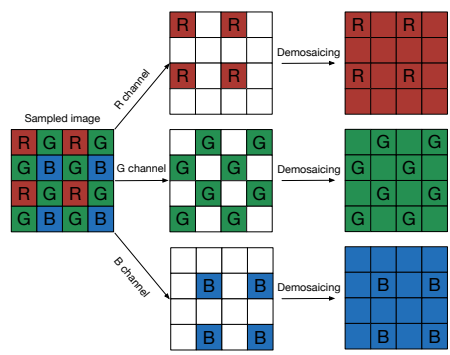


Figure 9.4: The process of CFA interpolation (demosaicing).

In consequence of cost considerations, most consumer digital cameras are equipped with only one imaging sensor and an associated color filter array (CFA). Only one color component passes through CFA is captured at each pixel and consequently forms a mosaic-like monochrome image, as shown in Fig. 9.4. The missing components have to be interpolated based on the captured components to recover the full-color image. The process of CFA interpolation is also known as demosaicing (Fig. 9.4), which will introduce specific inter-pixel and inter-channel correlations in the recovered full-color image. Therefore, by detecting the specific patterns introduced by demosaicing, we are able to infer the provenance of images.

Demosaicing has been extensively exploited for forensic purposes. Popescu and Farid [21] modeled the correlation between each pixel $I(x,y)$ and its neighboring pixels using a linear model

$$I(x,y) = \sum_{i=-N}^{N} \alpha_i I(x+\Delta x_i, y+\Delta y_i) + n(x,y), \tag{9.3}$$

where $n(x,y)$ is the modeling error, $(2N+1)\times(2N+1)$ is the size of the neighborhood, $\{\alpha_i | -N \le i \le N\}$ are the interpolation coefficients, and $(\Delta x_i, \Delta y_i)$ is the offset of the ith pixel within the neighborhood. They assumed that image tampering will likely destroy the inter-pixel correlations or produce inconsistent correlations. To reveal the potential tampering, they employed the expectation/maximization (EM) algorithm to simultaneously estimate the interpolation coefficients α_i and the probability map p indicating how likely one pixel conforms to the neighboring correlation characterized by Eq. (9.3). An image region without the presence of periodic pattern in the probability map p is considered as tampered, otherwise it is non-tampered. They also observed that the magnitude spectrum of the probability map p varies from one demosaicing algorithm to another. Motivated by this observation, Bayram *et al.* [22] used the peak locations and magnitudes in the DFT spectrum of the probability map p, along with the interpolation coefficients α, as features to differentiate camera models. With the LibSVM classifier [23] and the sequential forward floating search (SFFS) algorithm [24] for feature selection, they reported an average classification accuracy of 83.3% on three cameras of different brands. The accuracy on the same dataset was improved to 96% when the smooth and non-smooth image regions were handled differently [25].

In [26], Long and Huang formulated the Mean Square Error of $n(x,y)$ across the entire image in a quadratic form

$$\frac{1}{WH}\sum_{x=1}^{W}\sum_{y=1}^{H} |n(x,y)|^2 = X^T A X, \tag{9.4}$$

where $X = \{\alpha_{-N}, ..., \alpha_N, \alpha_{N+1}\}^T$ and

$$A(i,j) = \frac{1}{WH}\sum_{x=1}^{W}\sum_{y=1}^{H} I(x+\Delta x_i, y+\Delta y_i) I(x+\Delta x_j, y+\Delta y_j), -N \le i,j \le N,$$

with $\alpha_{N+1} = -1$ and $\Delta x_{N+1} = \Delta y_{N+1} = 0$. Instead of using the interpolation coefficients, they represented A in a 13×13 neighborhood as a 169-dimensional feature,

the dimension of which will be reduced to 15 using the principal component analysis (PCA) [27]. Instead of using the SVM classifier like in [22, 25], a 3-layer feed-forward neural network was trained using the 15-dimensional features for classifying different camera models. Almost perfect results were reported on the uncompressed images produced by digital cameras (four cameras and one class of cartoon images). But their algorithm tends to be sensitive to JPEG compression and median filtering.

Considering that the demosaicing algorithm may handle different image regions differently, Swaminathan *et al.* [28, 29] firstly divided the image regions into three categories:

- Region containing pixels with a significant horizontal gradient,
- Region containing pixels with a significant vertical gradient,
- Region that is mostly smooth.

Then they estimated the interpolation coefficients in each of the three regions by minimizing the approximation error over 36 CFA configurations. The estimation was performed in a 7×7 neighborhood in each of the three color channels (red, green, and blue), and thus results in $7 \times 7 \times 3 \times 3 = 441$ coefficients per image. Theses coefficients serve as 441-dimensional features that are fed into a probabilistic SVM classifier for training and classifying images of unknown provenance. An average classification rate of 90% was reported for 9 cameras of different brands, but it drops to 86% on a larger dataset consisting of 19 cameras of different models [29]. In the latter case, the classification errors were largely attributed to the ambiguities among cameras of the same brand.

Most advanced demosaicing algorithms often exploit the color difference and inevitably introduce strong inter-channel dependencies. However, the aforementioned algorithms only consider the inter-pixel correlations in the same color channel but ignore the inter-channel correlations. Cao and Cot [30] found that demosaicing is equivalent to estimating the second-order derivatives of neighboring pixels. Thus, they modeled the pixel dependencies using a partial second-order derivative correlation formula, which takes both the intra-channel and inter-channel correlations into consideration. Additionally, they proposed an expectation/maximization reverse classification (EMRC) algorithm to simultaneously classify the pixels demosaiced by the same formula into one of 16 demosaicing categories and estimate the interpolation coefficients representing the underlying demosaicing formulas. Based on the outcomes of the EMRC algorithm, a total of 1536 features were computed for each image. Similar to the work in [22, 25], the SFSS algorithm and the LibSVM classifier were jointly used to classify the images of unknown source. With 250 features selected by SFFS, an average accuracy of 97.5% was achieved over a dataset of 14 cameras, some of which are of the same brand. The leading-edge performance was also confirmed by the results on a dataset of 15 mobile cameras [31]. But as expected, the algorithm tends to confuse the cameras of the same model due to the identical in-camera demosaicing algorithm [31].

9.4.3 Camera Response Function

In digital cameras, the camera response function (CRF), which is generally a non-linear mapping, is used to transform the wide-ranging *irradiance* (i.e., the output of demosaicing) to a limited range of measurable image intensities. The principle of using CRF for image provenance inference is that cameras of the same model are expected to employ the same CRF. By adopting the CRF estimation algorithm in [32], Lin *et al.* [33] defined three features to measure the properties and consistencies of the CRFs recovered from the CRFs of different color channels. Positive samples (i.e., the normal CRFs collected from the DoRF database [34]) and negative samples (i.e., the abnormal CRFs estimated from forged images) are used to train an SVM predictor, which will be used to predict a confidence indicating the normality of the CRFs estimated from an image in question. The effectiveness of the algorithms was validated via comparison experiments on a few forged and non-forged images.

Hsu and Chang [35–37] proposed a CRF-based image splicing detection algorithm. They modeled the CRF using the Generalized Gamma Curve Model (CGCM) and estimated the parameters of CGCM using geometry invariants (GIs) calculated from locally planar irradiance points (LPIPs) in an image [38]. By automatically segmenting an image, they first obtained the suspect spliced regions and the boundary segments between neighboring regions. If the image is authentic, the GIs in the region on one side of a boundary segment is expected to fit well to the CRF estimated from the region on the other side of the boundary segment. Otherwise, if the regions on two sides of a boundary segment are from different cameras, it is expected to see large cross fitting errors. Therefore, they calculated 20-dimensional features for each boundary segment to measure the cross fitting errors as well as the fitness of CRF estimation, and classified the segments as authentic or spliced using an SVM classifier. They reported an image-level classification accuracy of 70% precision and 70% recall rate, on a dataset consisting of 180 spliced and 183 authentic images taken by four cameras.

While early works focused on exploiting the abnormality or inconsistency of CRFs from different sources for detecting image manipulations, the concept of camera CRF signature for distinguishing different camera models was introduced later in [39, 40]. The authors extended their CRF estimation algorithm [38] and defined a CRF signature as the histogram of the fitness scores of selected points with regard to the estimated CRF. Visual examination on the CRF signatures of four different camera models shows that cameras of different models are prone to have CRF signatures of different shapes, and the CRF signatures extracted from images with the same CRF tend to be consistent. However, the above-mentioned experiments related to CRF, for either image manipulation detection or image source identification, were conducted on datasets involving only a few cameras. Further studies on the distinctiveness of CRF over a large set of different camera models are yet to be conducted.

9.4.4 Quantization Table

In the last step of the imaging pipeline, most digital cameras export images in JPEG format. This lossy compression standard employs quantization tables to control the desired amount of compression. Although the Independent JPEG Group (IJG) recommended using the standard quantization tables, the JPEG users (e.g., camera manufacturers and computer programs) are free to specify their own quantization tables. In fact, the majority of cameras employ a different set of quantization tables. Therefore, comparing the quantization tables of different images offers a simple way to distinguish different image sources.

The idea of using JPEG quantization tables for camera ballistics was first proposed by Farid in [41], where an initial investigation on 204 cameras revealed that 62 (30.4%) out of the 204 cameras had a unique quantization table. While in the remaining cameras, not only the cameras from the same manufacturer may share the same quantization table, but even different makes and models are likely to have identical quantization tables. His followup study on a larger database [42] shows that the distinctiveness of quantization table can be even lower, with only 517 (5.1%) out of 10153 entries having a unique table. But an independent investigation carried out by Sorell [43] leads to a different interpretation. He found that, among the 330 distinct quantization tables extracted from 5485 images, over 92% of them are unique to one camera series. Furthermore, even after recompression of an image, residual artifacts of double compression continue to provide useful information for source camera identification.

Given the above attempts to discriminate image source via quantization tables, there is no denying that the quantization table is a reasonable discriminator between model series and effective at narrowing down the source of an image to a smaller set of possible cameras, but apparently it is insufficient to uniquely identify the image source. Besides, as pointed out in [43], the distinctiveness of quantization tables can be obscured by the smaller valued quantization tables adopted in high quality cameras for acquiring higher quality images.

9.4.5 Image Thumbnail

Another in-camera operation is the generation of an image thumbnail, which is a thumbnail sized version of the full resolution image and typically stored in the header of a JPEG image. A thumbnail is used to quickly preview the image without loading and displaying the full-sized image. The creation of a thumbnail involves a series of operations including cropping, blurring, down-sampling, sharpen, contrast and brightness adjustment, and JPEG compression. The parameters of these operations vary across camera brands or even models, and thus can be used to identify the source device of an image. Experimental results on 1514 images covering 142 cameras of different makes and models show that 40.8% of the cameras can be uniquely identified by using the thumbnail parameters [44].

To increase the accuracy of camera identification, one can simply incorporate more information. For example, by jointly using quantization tables, thumbnails

parameters and full resolution image size, the percentage of cameras that can be uniquely identified increases from 40.8% to 72.2% [11]. The results on a much larger database [11] show that a 576-valued camera signature, consisting of the information from quantization tables, Huffman codes, thumbnail parameters and EXIF metadata parameters, is capable of uniquely identifying 69.1% of 9163 camera configurations. It should be noted that the percentage of cameras that can be uniquely identified varies from one database to another, and also depends on what "fingerprints" are used for identification. But any of the device fingerprints we have discussed so far is only sufficient for brand-level or model-level image provenance inference. So even involving all of the aforementioned fingerprints, ambiguities still abound in identifying the individual devices of the same brand or model.

9.5 Sensor Pattern Noise

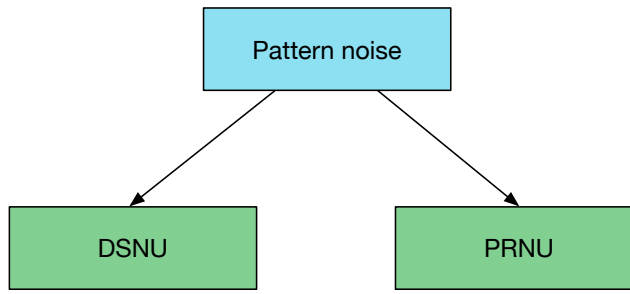


Figure 9.5: Pattern noise of imaging sensors.

A promising method for uniquely distinguishing individual devices is based on sensor pattern noise (SPN) [45]. As shown in Fig. 9.5, pattern noise consists of two main components. One is the fixed pattern noise (FPN) (or dark current noise as it is more commonly referred to as), which is the pixel-to-pixel differences when the sensor array is not exposed to light. The dominant component in SPN is the photo response non-uniformity (PRNU) noise. It is primarily resulted from the variation among pixels in their sensitivity to light, which is caused by the manufacturing imperfections and the inhomogeneity of silicon wafers during the sensor manufacturing process [45]. It has attracted much attention from researchers in the past decade because of its desired characteristics:

1. **Universality**. Every imaging sensor exhibits SPN, therefore the methods based on SPN are widely applicable to any device equipped with an imaging sensor.

2. **Stability**. SPN is not subject to the influence of environmental conditions, such as temperature and humidity, and essentially time-independent.

3. **Robustness**. SPN is robust to common image processing operations, such as JPEG compression, gamma correction, and image filtering.

4. **Uniqueness**. SPN can be considered as unique to each sensor because of the large number of pixels of the sensor and the randomness of SPN.

Therefore, SPN is considered as the fingerprint of imaging devices and has been widely and successfully applied image provenance inference. In the following subsections, we will introduce the estimation of SPN, and its use in source camera identification, device linking, source-oriented image clustering and image forgery detection.

9.5.1 Estimation of SPN

The SPN of a device can be estimated from the noise residuals extracted from a collection of images acquired by the device. The noise residue is defined as the difference between the original image $\mathbf{I}$ and its denoised version $\hat{\mathbf{I}}^{(0)}$ [46]:

$$\begin{aligned}\mathbf{W} =& \mathbf{I} - \hat{\mathbf{I}}^{(0)} && (9.5)\\ =& (\mathbf{1}+\mathbf{K})\mathbf{I}^{(0)} + \boldsymbol{\Theta} - \hat{\mathbf{I}}^{(0)} && (9.6)\\ =& \mathbf{I}\mathbf{K} + \mathbf{I}^{(0)} - \hat{\mathbf{I}}^{(0)} + (\mathbf{I}^{(0)} - \mathbf{I})\mathbf{K} + \boldsymbol{\Theta} && (9.7)\\ =& \mathbf{I}\mathbf{K} + \boldsymbol{\Xi}, && (9.8)\end{aligned}$$

where $\hat{\mathbf{I}}^{(0)}$ is the estimation of the noise-free image $\mathbf{I}^{(0)}$ and can be obtained by applying a denoising filter $F(\cdot)$ to $\mathbf{I}$, i.e., $\hat{\mathbf{I}}^{(0)} = F(\mathbf{I})$, $\mathbf{K}$ is the noise-like multiplicative factor responsible for PRNU noise, $\boldsymbol{\Theta}$ stands for a complex of independent random noise components containing the interferences from image content and other noises, and $\boldsymbol{\Xi}$ is the sum of $\boldsymbol{\Theta}$ and the two additional terms $\mathbf{I}^{(0)} - \hat{\mathbf{I}}^{(0)}$ and $(\mathbf{I}^{(0)} - \mathbf{I})\mathbf{K}$. $\mathbf{I}\mathbf{K}$ can be reasonably assumed to be independent of $\boldsymbol{\Xi}$ as the term $(\mathbf{I}^{(0)} - \mathbf{I})\mathbf{K}$ in $\boldsymbol{\Xi}$ is very weak [46].

The estimation of SPN is usually referred to as the *reference* SPN (RSPN), which is considered as the unique fingerprint of a source device. It can be obtained by averaging the noise residuals of N images taken by the source device:

$$\mathbf{R} = \frac{1}{N}\sum_{k=1}^{N}\mathbf{W}_k, \tag{9.9}$$

where $\mathbf{W}_k$ represents the noise residual of the kth image. Alternatively, the PRNU term $\mathbf{K}$ can be explicitly estimated through a maximum likelihood estimate (MLE) method [46]:

$$\hat{\mathbf{K}} = \frac{\sum_{k=1}^{N}\mathbf{W}_k\mathbf{I}_k}{\sum_{k=1}^{N}\mathbf{I}_k^2}. \tag{9.10}$$

By comparison, Eq. (9.9) not only considers the PRNU noise but also implicitly includes the FPN in $\mathbf{R}$. It is worth mentioning that the FPN is not as stable as PRNU and may have been removed by dark-frame subtraction in device, but it facilitates

the image provenance inference if it remains in the image. To better estimate the SPN, the image intensity I_k should be as high as possible but not saturated because of the multiplicative nature of the PRNU noise IK [46]. Also note that in Eq. (9.8) the smaller the variance of undesired signal $\boldsymbol{\Xi}$, the more accurate the estimation in Eq. (9.9) and Eq. (9.10). Therefore, images of bright and smooth scenes, such as blue sky and flat field (i.e., intensities are approximately constant) images, are commonly used for SPN estimation. In the case of flat field images, there is no much difference between the simple averaging method in Eq. (9.9) and the MLE method in Eq. (9.10). Unless otherwise stated in the rest of this chapter, RSPN refers to the estimation of SPN using the simple averaging method in Eq. (9.9).

9.5.2 Source Device Identification

As shown in Fig. 9.6, to identify the source device among a set of candidate devices $\mathscr{C}$ for a *test* image of unknown source, the typical process is to calculate the normalized cross-correlation (NCC) between the noise residual $\mathbf{W}$ of the test image and the RSPN $\mathbf{R}_c$ of each device $c \in \mathscr{C}$ [45]:

$$\rho(\mathbf{R}_c, \mathbf{W}) = \frac{\sum_{k,l}(\mathbf{W}(k,l) - \overline{\mathbf{W}}(k,l))(\mathbf{R}_c(k,l) - \overline{\mathbf{R}}_c(k,l))}{\sqrt{\sum_{k,l}(\mathbf{W}(k,l) - \overline{\mathbf{W}}(k,l))^2}\sqrt{\sum_{k,l}(\mathbf{R}_c(k,l) - \overline{\mathbf{R}}_c(k,l))^2}}, \tag{9.11}$$

where the mean value is denoted with a bar and $||\cdot||$ is the $L2$ norm. The test image is deemed to be taken by the camera c^* with the maximal NCC value that is greater than a predefined threshold τ:

$$c^* = \underset{c \in \mathscr{C}}{\operatorname{argmax}}\{\rho(\mathbf{R}_c, \mathbf{W})\}, \rho(\mathbf{R}_{c^*}, \mathbf{W}) > \tau, \tag{9.12}$$

where τ is usually determined by Neyman-Pearson criterion [46]. Although more advanced detection statistics such as the peak-to-correlation energy (PCE) [47] and the correlation over circular cross-correlation norm (CCN) [48] have been proposed to improve the identification performance, NCC is still the most widely adopted SPN similarity measurement probably because of its simplicity.

SPN has demonstrated great promise in discriminating individual devices. Large-scale tests on millions of images spanning 6896 individual cameras covering 150 models show a very high detection rate 97.6% at a false positive rate as low as 2.4×10^{-5} [49]. The great potential of SPN has attracted much attention from researchers, and many efforts have been devoted to improving the performance of SPN-based source device identification. A feasible direction of improvement is to adopt more advanced denoising filters because the performance of the denoising filter $F(\cdot)$ has a direct impact on the quality of the noise residual. The basic and probably the most prevailing denoising filter for estimating the SPN is based on the Mihcak filter [50]. It works by calculating the fourth-level wavelet decomposition of the image and applying the Wiener filter in the high-frequency subbands in each of the four levels. Chierchia *et al.* [51] proposed to use a non-local denoising filter, block-matching and 3D filtering (BM3D) [52], which works by grouping 2D similar image

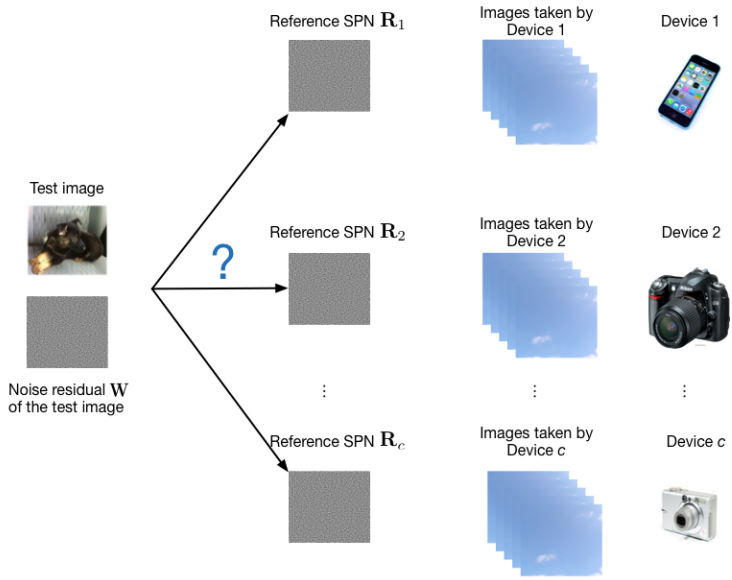


Figure 9.6: Source device identification based on SPN.

patches found across the entire image into 3D arrays and collectively filtering the grouped image blocks. The sparseness of the representation due to the similarity between the grouped blocks makes it capable of better separating the true signal and noise. The results in [51] show that BM3D better prevents the image scene from propagating to the noise residual than the Michak filter. Another denoising filter, edge adaptive SPN predictor based on context adaptive interpolation (PCAI) [53, 54], was proposed to suppress the effect of scene edges. It first predicts the value of a pixel from its neighboring pixels, then a Wiener filter is applied to the difference between the predicted image and the original image to obtained the noise residual. Because the prediction of pixel values is edge-aware, the noise residual obtained with PCAI is expected to have less scene details than that obtained with the Michak filter.

Another line of research is dedicated to selecting or weighting the components in the noise residual $\mathbf{W}$ aiming to strengthen the PRNU signal $\mathbf{IK}$ and suppress the undesired signal $\boldsymbol{\Xi}$ in Eq. (9.8). Li [55] proposed five models to attenuate scene details by assigning less significant weighting factors to the strong components of SPN in the wavelet domain. The underlying rationale is that the stronger a component in $\mathbf{W}$ is, the more likely it is associated with strong scene details, and thus the less trustworthy the component should be. Lin and Li [56] further improved the quality of $\mathbf{W}$ by abandoning the components that have been severely contaminated by denoising errors. As mentioned in the last subsection, SPN is better preserved in images with high intensity and low textured scenes. For this reason, McCloskey [57] suggested giving a smaller weight to the pixels with a larger local gradient. A

more sophisticated weighting scheme in [58] predicts the correlations between the blocks in noise residual $\mathbf{W}$ and the corresponding blocks in the RSPN $\mathbf{R}$ using image intensity and texture features. A block with a larger predicted correlation is expected to contain SPN of higher quality, so a larger weight is assigned to the center pixel of the block. While all the aforementioned weighting schemes deal with the noise residual $\mathbf{W}$ of the test image, some other works shifted the focus to the RSPN $\mathbf{R}$. Hu *et al.* [59] assumed that the larger components of $\mathbf{R}$ are more reliable while the smaller components are more sensitive to random noise. So they proposed to involve only a certain percent (e.g., 10%) largest components of $\mathbf{R}$ into the calculation of correlation. Li and Li [60] proposed an estimator to construct a reliable RSPN from a limited number of images. Specifically, each image $\mathbf{I}_k$ and the corresponding noise residual $\mathbf{W}_k$ are segmented into non-overlapping blocks. The quality of each block is then measured and sorted based on the entropy and average intensity of the block. The block with a higher ranking (i.e., higher quality) is assigned a larger weight. Finally, the RSPN in different block locations is estimated as the weighted average of the noise residuals in the same block location. A similar RSPN estimator was proposed in [61], where the equal weighting factor $1/N$ in Eq. (9.9) is replaced with a new weighting factor related with the variances of the undesirable noise in the noise residual $\mathbf{W}_k$.

Interestingly, while some of the processing components in the image acquisition pipeline introduce specific patterns or characteristics that are useful for identifying the source device, they may become the *interference sources* for the accurate estimation of SPN. For this reason, some works have been proposed to alleviate the influence of the "interferences" introduced during the image acquisition process. For example, the artificially interpolated color samples obtained through demosaicing are not physically captured by the sensor, thus the SPN components extracted from the artificial samples are expected to be less reliable than those extracted from the physical samples. Based on this assumption, Li and Li [62] proposed a Couple-Decoupled PRNU (CD-PRNU) extraction method to prevent the interpolation noise from propagating into the PRNU noise extracted from physical components. They first decomposed each color channel into four sub-images and extracted the noise residual from each sub-image. Finally, they assembled the noise residuals of the sub-images to obtain the CD-PRNU. Another source of interfering is the in-device lens-distortion correction, which allows users to take high-quality photos at a wide range of zoom. The lens-distortion correction desynchronizes the pixel-to-pixel correspondence between images taken at two different focal lengths and thus leads to a low accuracy for SPN-based source device identification. To reestablish synchronization between an image and the RSPN, Goljan and Fridrich [63] adopted the barrel distortion model in [18] and search for its parameter to maximize the detection statistic between the noise residual and the RSPN. In [64], they extended their method to make it work for single images (i.e., without the RSPN) by searching for a maximum energy of the linear pattern [46] introduced into the image prior to lens distortion correction.

The processing components in the image acquisition pipeline not only inflict distortion but also introduce non-unique artifacts (NUAs) in the noise residual. These

NUAs are shared among the devices with the same or similar in-camera processing procedures. The unwanted artifacts including the demosaicing artifacts, JPGE block artifacts, and the diagonal artifacts reported in [65], may give rise to false positives[1] and thus should be suppressed for improving the reliability for SPN-based device identification. Chen *et al.* [46] proposed two preprocessing operations to suppress the NUAs in RSPN. One operation is zero-meaning (ZM) operation, which removes the linear pattern in RSPN by subtracting the column average from each pixel in the column and subtracting the row average from every pixel in the row. The other operation is Wiener filtering (WF) in the frequency domain, which attenuates the periodic artifacts in RSPN. Kang *et al.* [48] suggested only keeping the phase components of the noise residual when constructing the RSPN. The underlying rationale is that the SPN is usually modeled as an additive white Gaussian noise (AWGN) in its estimation process, so it is reasonable to assume that the RSPN is a white noise signal with flat frequency spectrum to facilitate the removal of the contamination in the frequency domain [48]. Only keeping the phase components whitens the noise residual in the frequency domain and helps to remove the periodic artifacts. In view of the fact that the periodic artifacts manifest themselves as peaks in the DFT spectrum, Lin and Li [66] proposed a spectrum equalization algorithm (SEA) to detect and suppress the peaks in the magnitude spectrum of RSPN.

9.5.3 Device Linking

Another important application of SPN is device linking. As the name suggests, it is about linking the images acquired by the same device. But in the scenario of device linking, the source device or any other image from it is not available. In a typical case, we would like to know whether or not two given images came from the same camera, as shown in Fig. 9.7. Device linking is particularly useful in the analysis of digital evidence in law enforcement when the source device is unavailable to the forensic investigators. One such example is the forensic investigation of on-line child abuse, where the criminals record moments of the ongoing crime by taking videos or images and share them over the Internet. These criminal recordings (not necessarily taken by the same device) are often accessible to the forensic investigators. By matching the recordings to those posted in social networks accounts that belong to suspected persons, the forensic investigators are able to find out the criminals [67]. Note that because the absence of the source device and the images taken by it prohibits the acquisition of a reliable RSPN, device linking can only be carried out based on the noise residual from each image, which may have been severely contaminated by other SPN-irrelevant interferences. Therefore, SPN-based device linking is a more challenging problem than the SPN-based source device identification.

The goal of device linking can be achieved by simply calculating the similarity (e.g., NCC) between each pair of the noise residuals extracted from the images under investigation and comparing it with a predefined threshold. But the images under investigation may differ in size, e.g., when one or several of them have been cropped.

[1] The SPN signals estimated from two devices may be slightly correlated due to the presence of NUAs.

In view of this problem, Goljan *et al.* [68] recommended padding the images of different sizes with zeros and calculating the normalized circular cross-correlation [47], rather than the NCC, between each pair of the noise residuals of the images. If the ratio of the primary peak to the secondary peak (PSR) of the normalized circular cross-correlation is higher than a threshold determined by Neyman-Pearson criterion, the images are believed to be from the same device. Unlike the abundant research in enhancing the performance of SPN-based source device identification, there have been few studies on SPN-based device linking in spite of its many potential applications. As far as we know, among the methods aiming to enhance the performance of source device identification, only Li's enhancer [55] has been applied to boost the performance of device linking. This is partially because that the absence of RSPN invalidates the enhancing methods that attempt to improve the quality of RSPN. Further studies are still needed to verify the effectiveness of the enhancing methods in section 9.5.2 for device linking.

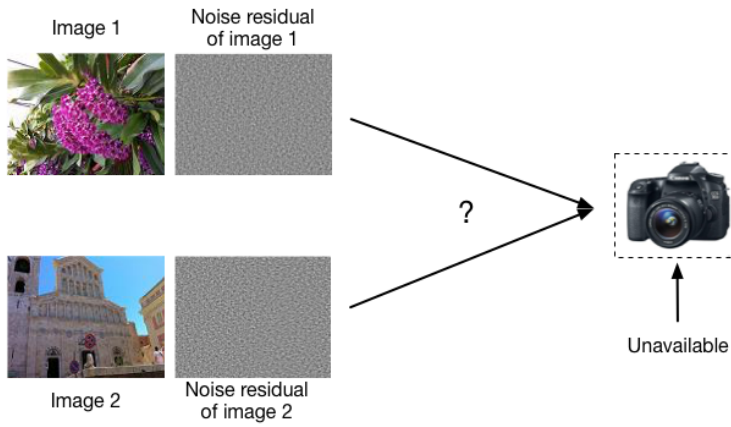


Figure 9.7: Device linking based on SPN.

9.5.4 Source-Oriented Image Clustering

There are circumstances where forensic investigators want to cluster a set of images taken by an unknown number of devices into a number of groups, such that the images in each group are acquired by the same device. Taking the aforementioned on-line child abuse as an example, if the forensic investigators can cluster a set of criminal images into groups, each including the images taken by the same device, they are able to link different crime scenes together and may obtain extra information from the grouped images (e.g., the images appearing in different social network accounts are associated to the same criminal). We refer to this task as the source-oriented image clustering. Since SPN is considered as the unique fingerprint of a device, source-oriented image clustering can be accomplished by extracting SPN from each image and then clustering the images based on the similarities between

corresponding SPNs. Similar to SPN-based device linking, we do not have the access to the source devices or the reference SPNs for source-oriented image clustering, so only the SPNs (i.e., noise residuals) extracted from single images are available. Source-oriented image clustering is seemingly similar to but actually differs from device linking. Device linking checks whether a limited number of (typically two) images are taken by the same device, so it involves only one device though the device itself is not available. While for source-oriented image clustering, both the number of devices and the number of images taken by each device are unknown. It may involve a large set of images, which makes the pairwise comparison in device linking computationally prohibitive for source-oriented image clustering. Moreover, to obtain accurate clustering, the dimension of SPN has to be very large, e.g., 512×512 pixels or above. The high dimension of SPN will impose a heavy burden on computation. All these difficulties make source-oriented image clustering much more challenging than device linking.

Bloy [69] presented a heuristic algorithm for clustering images iteratively based on SPN (i.e., noise residual), with the aim of forming one cluster in each iteration. To form a cluster, the algorithm randomly selects pairs of images until a pair is found to have an SPN correlation (i.e., NCC) higher than a threshold. The average SPN of the image pair, which serves as the cluster centroid, is correlated with the SPN of each remaining image. If one correlation exceeds a threshold that adaptively increases with the number of SPNs (images) in the cluster, the corresponding image is assigned to the cluster and the SPN of the image is averaged into the cluster centroid. When the number of images in the cluster reaches 50, the algorithm stops updating the centroid but continues to add more similar images to the cluster until the entire dataset has been exhausted. Once a cluster is formed, the algorithm starts another iteration to form a new cluster until no further clustering is possible. The cluster centroid actually plays the same role as the RSPN in Eq. (9.9) and becomes more reliable as more SPNs are averaged into it. Note that the threshold used for determining whether or not one image belongs to the cluster should be able to well characterize the change of correlation after updating the centroid. However, the adaptive threshold in [69] was obtained by fitting a quadratic threshold curve based on the SPN correlations calculated from images taken by four cameras. It does not generalize well across different cameras and results in unsatisfactory clustering results. Moreover, one image (or its SPN if all the SPNs have been extracted beforehand) will be repeatedly loaded into the RAM until it has been clustered, which incurs extra I/O cost and makes the algorithm computationally infeasible for large-scale image databases.

Li [70] proposed to cluster a subset of images (*training set*) randomly chosen from the entire database and use the clustering results of the training set to classify the remaining images. Prior to the actual clustering, SPNs of the training set are extracted for constructing a pairwise similarity matrix, with the element at index (i, j) being the NCC between SPNs W_i and W_j. Based on the pairwise similarity matrix, a reference similarity and a membership committee are set up for each image to estimate the likelihood probability of assigning each class label to the corresponding image. The class label of each image is updated as the one with the highest likelihood probability in its membership committee. The clustering process terminates

when there are no label changes in two consecutive iterations. Finally, the remaining images are attributed to their closest clusters identified in the training set. In spite of the good performance (an overall error rate of 1.444% using SPNs of 512×512 pixels), this clustering algorithm is very slow because the calculation of the likelihood probability involves all the members and their class labels in the membership committee. The time complexity is nearly $\mathcal{O}(N^3)$ in the first iteration, where N is the number of images in the training set. For large-scale image databases, the size of the training set has to be sufficiently large to well represent the entire database, so the algorithm becomes computationally prohibitive for large-scale image databases.

Liu *et al.* [71] formulated the source-oriented image clustering as a weighted undirected graph partitioning problem, where each image is considered as a vertex in the graph and the weight of an edge is the SPN similarity (i.e., NCC) between the two images linked by the edge. Instead of a fully connected graph, a sparse k-nearest graph is constructed to avoid calculating the similarity of every pair images. An m-class spectral clustering algorithm [72] is then employed on the k-nearest graph to partition the vertices (images) into m clusters. The m-class spectral clustering algorithm has a time complexity of $\mathcal{O}(N^{\frac{3}{2}}m + Nm^2)$, so it is more efficient than Li's algorithm [55] when $N \gg m$. But the spectral clustering algorithm requires an input of the cluster number m, which is unknown to the user. To determine the optimal cluster number, the same spectral clustering algorithm needs to be repeated for different value of m until the smallest size of the resultant clusters equals 1, i.e., one singleton cluster is generated. However, it is easy to form singleton clusters because some SPNs may have been severely contaminated by interferences, such as scene details [55] and CFA interpolation artifacts [62]. So the feasibility of such manner of determining the optimal cluster number is still an issue for source-oriented image clustering based on SPN.

Caldelli [73] proposed a hierarchical clustering algorithm for source-oriented image clustering. Similar to [70], only a random subset (training set) of the whole dataset is used for clustering, followed by a classification stage for the remaining images. Initially considering each image as a cluster, the algorithm first calculates the pairwise similarity matrix of the SPNs in the training set. It then merges the two most similar clusters into one and updates the similarity matrix by replacing the corresponding two rows and columns with the similarities between the merged cluster and all the other clusters. After the update, a silhouette coefficient, which measures the separation among clusters and the cohesion within each cluster, is calculated for each SPN. The silhouette coefficients are averaged to give a global measure of the aptness of the current partition. This procedure repeats until all the images have been merged into one cluster. Upon completion of the clustering, the partition corresponding to the highest aptness is taken as the optimal partition. Villalba *et al.* [74] proposed a similar algorithm for smartphone image clustering. Its difference from [73] is that the calculation of the silhouette coefficient is performed for each cluster rather than for each fingerprint and only the separation to the nearest neighboring cluster is measured. As reported in [73], with comparable accuracy, the hierarchical clustering based algorithm is faster than [70]. But the time complexity $\mathcal{O}(N^2 \log N)$ of the hierarchical clustering still too high for large-scale image databases.

It can be seen that the large-scale source-oriented image clustering problem cannot be well resolved by the above algorithms due to the large-scale number of images and the high dimension of SPNs. To alleviate the problem, the algorithms in [55] and [73] first cluster a training set randomly sampled from the entire database and classify the remaining images based on the clustering results of the training set. They work well if the training set can sufficiently represent the entire database, i.e., the training set includes a portion of images taken by all or most of the devices appearing in the entire database. However, sometimes the Number of Classes (i.e., the number of devices) is much higher than the average Size of Class (i.e., the number of images acquired by each device), which was referred to as the $NC \gg SC$ problem in [75]. The $NC \gg SC$ problem makes it difficult, if not impossible, to form a training set at random that can sufficiently represent the entire population.

To overcome these challenges, Lin and Li [75] proposed a clustering framework capable of handling large-scale image databases. By taking advantage of dimension reduction and the inherent sparseness of the pairwise similarity matrix, the algorithm first roughly but efficiently partitions the entire database into small subsets, with larger classes having a higher chance to be partitioned into the same subset. It then clusters each subset using the Markov cluster algorithm [76] to produce many small but highly pure sub-clusters, with each of them represented by an SPN centroid, the cluster size and a cluster quality coefficient (calculated from the pairwise correlations of SPNs within the sub-cluster). If the similarity between the SPN centroids of two sub-clusters is higher than an adaptive threshold, the two sub-clusters will be merged and the SPN centroid of the merged cluster will be updated at the same time. It is worth mentioning that the adaptive threshold in [75] takes both the size and the quality of clusters into consideration and thus is more accurate than the threshold in [69], which only considers the cluster size. The centroids of the merged clusters will be used to attract the remaining images in the database, but unlike the classification stage in [70] and [73], an adaptive threshold is used to reduce the false attributions[2], and the centroid and the quality of the clusters will be updated accordingly after attracting a certain number of images. The above procedures are repeated until no more notable clusters can be discovered. Because the algorithm allows larger classes to be clustered preferentially, the majority of images can be clustered in the first few iterations. The results on the 15840 images in the Dresden image database [77] show that the algorithm is much more efficient than the algorithms in [55, 71, 73] and delivers a high level of clustering quality using SPNs of 1024×1024 pixels, with a precision rate of 99% and an F1-measure of 68%. It also demonstrates a high capability of solving the $NC \gg SC$ problem. On synthetic datasets consisting of 50000 images, about 92% of classes are discovered when $NC = 1250$ and $SC = 40$, and more than 76% of classes are discovered when $NC = 2500$ and $SC = 20$.

[2]False attribution happens when one image is attributed to the cluster with the highest similarity, but actually it does not belong to the cluster and their similarity is still very low.

9.5.5 *Image Forgery Detection*

Detecting image forgeries is an interesting while very challenging task due to the variety of image manipulations a user can perform with increasingly powerful image editing software. SPN exists in every *original* image taken by the source device, while the image forgery may damage or remove the SPN signal that is supposed to present in the forged regions. Therefore, when the RSPN of the source device is available, image forgeries can be exposed by detecting the absence of the SPN in suspect regions. Since SPN is the intrinsic fingerprint of the source device and not associated with any type of image forgeries applied after the image acquisition, techniques based on SPN can detect the image forgeries irrespective of the specific type of forgery. In addition, SPN is robust to some common image processing operations, such as JPEG compression, filtering, or gamma correction [45, 46]. These characteristics make SPN a promising tool for detecting image forgeries.

Lukas *et al.* [78] proposed two approaches, respectively, for detecting the forgeries in a selected Region of Interest (ROI) and for automatically identifying the forged areas of an image $\mathbf{I}$ captured by a device, whose RSPN $\mathbf{R}$ is first constructed as in Eq. (9.9). In the first approach, to calculate the statistical evidence that a suspect region Ω in $\mathbf{I}$ has been tampered with, a large set of image regions $\mathbf{Q}_k, k = 1,...,N$ of the same size and shape as Ω is collected either from the images taken by the same device (but from regions different from Ω), or from the images taken by other devices. These image regions are considered as "tampered" regions because the SPN presents in them are different from that presents in region Ω of the RSPN. The correlations $\rho(\mathbf{R}_\Omega, \mathbf{W}_k)$ are supposed to be subject to a generalized Gaussian distribution, where $\mathbf{R}_\Omega$ is the RSPN in the same region as Ω and $\mathbf{W}_k$ is the SPN (i.e., noise residual) extracted from $\mathbf{Q}_k$. The smaller the correlation between the RSPN and the noise residual in Ω, i.e., $\rho(\mathbf{R}_\Omega, \mathbf{W}_\Omega)$, the more likely Ω has been tampered with. By fitting the generalized Gaussian distributing using the correlations calculated from the "tampered" regions, the probability that Ω has been forged can be calculated as

$$p = 1 - \Phi(\rho(\mathbf{R}_\Omega, \mathbf{W}_\Omega)), \tag{9.13}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the estimated generalized Gaussian distribution. Ω is forged if $p > \alpha (= 10^{-3})$, and not forged otherwise.

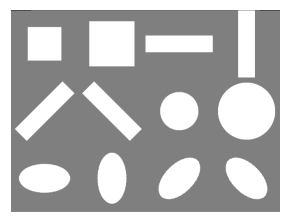


Figure 9.8: Window shapes used for automatic ROI detection in [78].

The second approach is capable of automatically identifying the forged area. To detect forgeries of different shapes, twelve detection blocks of different shapes and

sizes are prepared, as illustrated in Fig. 9.8. Each detection block $i \in \{1,...,12\}$ is moved across the image under investigation and the RSPN of its source device (overlapping approximately $50\% - 75\%$), and the correlation between the RSPN and the noise residual within the region covered by the detection block is calculated. For each block i, m regions with the smallest correlations, i.e., the m most likely forged regions, are selected (m was set to 8 in [78]). So there will be total of $m \times 12$ regions $\mathfrak{B}_k$ and their union $\mathfrak{B} = \bigcup_{k=1}^{m\times 12} \mathfrak{B}_k$ are initially identified as the forged regions. Then the algorithm tries to refine the initial result: For each pixel $q \in \mathfrak{B}$, if the number of selected regions covering q, i.e., $t(q) = |\{\mathfrak{B}_k | q \in \mathfrak{B}_k\}|$ is no higher than the median value of $t(q)$ across the initially detected regions $\mathfrak{B}$, i.e., $q \in \mathfrak{B}$, pixel q is corrected as non-forged.

The work in [78] was improved by Chen *et al.* in [46]. They modeled the SPN detection problem as a binary hypothesis testing problem:

$$\begin{cases} H_0 : \mathbf{W} = \mathbf{\Xi}, \\ H_1 : \mathbf{W} = \mathbf{R} + \mathbf{\Xi}, \end{cases} \tag{9.14}$$

where $\mathbf{W}$ is the noise residual extracted from the image region in question, $\mathbf{R}$ is the RSPN of the source device c, and $\mathbf{\Xi}$ is the combination of other independent interferences. The forgery detection at each pixel q is formulated as a hypothesis testing problem applied to a sliding block surrounding q. As illustrated in Fig. 9.9, a detection block[3] is sliding across the image and the test statistic $\rho_q = \rho(\mathbf{R}_q, \mathbf{W}_q)$ within the block is calculated, where $\mathbf{R}_q$ and $\mathbf{W}_q$ are the RSPN and the noise residual in the detection block centered at pixel q, respectively. This produces a correlation map $\boldsymbol{\rho}$, with the value at pixel q being the correlation ρ_q. Note that because SPN is pixel location sensitive, even if the forged region comes from the image taken by the same device c, it is still able to detect the forgery as long as the forged region does not exactly lie in the same position as in the image it comes from. To facilitate the decision-making, both the correlation distribution under hypothesis H_0, $p(x|H_0)$, and the correlation distribution under hypothesis H_1, $p(x|H_1)$ need to be estimated.

$p(x|H_0)$ can be easily estimated from the images taken by other devices, while the estimation of $p(x|H_1)$ is difficult, because the correlation is heavily affected by the image content (e.g., the correlation between SPNs tends to be higher for the images with brighter and smoother scenes) and is most likely to be over-fitting to the available images [46]. Therefore, instead of directly estimating $p(x|H_1)$ from correlations, the authors constructed a correlation predictor that maps the image features to the correlation value. Specifically, K image blocks of 128×128 pixels are cropped from several images taken by the source device. Four image features that affect the quality of SPN are extracted from each of the K image blocks: Image intensity feature f_I, texture feature f_T, signal flattening feature f_S, and texture-intensity feature f_E. Let $\boldsymbol{\rho}$ be the correlations between the noise residuals of the K image blocks and the reference SPN in the corresponding positions, and $\boldsymbol{f}_I$, $\boldsymbol{f}_T$, $\boldsymbol{f}_S$, and $\boldsymbol{f}_E$ be the corresponding K-dimensional feature vectors. $\boldsymbol{\rho}$ is modeled as a linear combination of the features and their second-order terms, i.e., $\boldsymbol{\rho} = \mathbf{H}\boldsymbol{\theta} + \mathbf{\Psi}$, where $\mathbf{\Psi}$ is the

[3] The size of the detection was set to 128×128 pixels in [46].

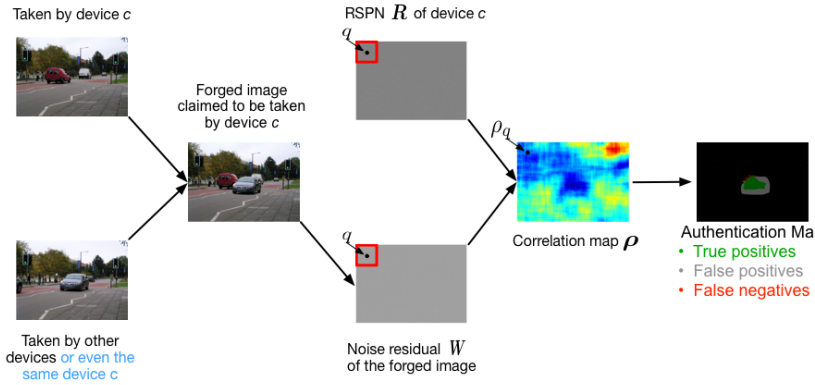


Figure 9.9: Image forgery detection based on SPN.

modeling noise, $\mathbf{H}$ is a $K \times 15$ matrix containing the features and their second-order terms, and $\boldsymbol{\theta}$ is the modeling coefficients to be determined. By applying the least square estimator (LSE), the estimated coefficients $\hat{\boldsymbol{\theta}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\boldsymbol{\rho}$ are obtained. So the expected correlation $\hat{\rho}$ of an unseen image block can be predicted based on the image features extracted from it:

$$\hat{\rho} = [1, f_I, f_T, f_S, f_E, ..., f_E f_E]\hat{\boldsymbol{\theta}}. \tag{9.15}$$

With the predicted correlation, the actual correlation ρ is modeled as a random variable following a generalized Gaussian distribution $G(\hat{\rho}, \sigma_1, \alpha_1)$, where the predicted correlation $\hat{\rho}$ is the mean, while the scale parameter σ_1 and the shape parameter α_1 can be estimated from the difference between the actual and predicted correlations of the K images blocks, i.e., $\boldsymbol{\rho} - \mathbf{H}\hat{\boldsymbol{\theta}}$. Then the decision is made for each pixel independently: If $\rho_q < t$, pixel q is labeled as forged. Here the threshold t is related with a constant false acceptance rate (CFAR) $\alpha (= 10^{-5})$:

$$\int_t^{\infty} p(x|H_0)\mathrm{d}x = \alpha. \tag{9.16}$$

Like in [79], we refer to this method as the constant false acceptance rate (CFAR) method in this chapter. However, for a highly textured, black or saturated block, even if it is authentic, its correlation still tends to be low due to the attenuation of SPN. So to reduce the false positives (i.e., labeling non-forged pixels as forged), a pixel q will be labeled as non-forged if $\int_{-\infty}^{t} p(x|H_1)\mathrm{d}x > \beta$, where β was set to 0.01 in [46]. The resulting binary map $\hat{\boldsymbol{u}} \in \{0,1\}^{M \times N}$ signifying the forged pixels (1 for forgery and 0 for genuine pixel) will be further dilated with a square 20×20 kernel to obtain the final result.

The CFAR method does not take into account the spatial dependencies exhibited by natural images but makes decisions independently for each pixel, which may generate inconsistent and fragmented binary map. To penalize the isolated points or the small disjoint regions and produce a smooth output, Chierchia *et al.* [79]

adopted the Bayesian approach and Markov random field (MRF) model to improve the detection result. This Bayesian-MRF method is based on the CFAR method but differs in both formulation and solution to the problem. It formulates the forgery detection as an optimization problem of finding the label map $\hat{\mathbf{u}} \in \{0,1\}^{M\times N}$ that has the maximum posterior probability given the prior information:

$$\hat{\mathbf{u}} = \underset{\mathbf{u}\in\{0,1\}^{M\times N}}{\operatorname{argmax}} \; p(\boldsymbol{\rho}|\mathbf{u},\hat{\boldsymbol{\rho}})p(\mathbf{u}), \tag{9.17}$$

where $M\times N$ is the image size, $\boldsymbol{\rho}$ is the actual correlations calculated in a block-wise manner (i.e., the correlation map in Fig. 9.9), and $\hat{\boldsymbol{\rho}}$ is the predicted (or expected) correlations given by the correlation predictor in Eq. (9.15). In the above equation, $p(\boldsymbol{\rho}|\mathbf{u},\hat{\boldsymbol{\rho}})$ is the conditional likelihood of observing $\boldsymbol{\rho}$, and $p(\mathbf{u})$ is the prior probability that takes into account the spatial dependencies of the pixels, which is modeled by the Markov random field model:

$$p(\mathbf{u}) = \frac{1}{Z} e^{-\sum_{c\in\mathscr{C}} V_c(\mathbf{u})}, \tag{9.18}$$

where Z is a normalizing constant, and $V_c(\mathbf{u})$ is the potential defined on cliques c (i.e., small groups of neighboring pixels). Only the single-site cliques and 4-connected two-site cliques are considered in [79]. By assuming the likelihood probability to be Gaussian under both hypotheses, with zero mean and variance σ_0^2 under hypothesis H_0, and mean $\hat{\rho}_i$ and variance σ_1^2 under hypothesis H_1 (obtained using the above-mentioned correlation predictor [46]), Eq. (9.17) is formulated as

$$\hat{\mathbf{u}} = \underset{\mathbf{u}\in\{0,1\}^{M\times N}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{M\times N} u_i \left[\frac{(\rho_i-\hat{\rho}_i)^2}{2\sigma_1^2} - \frac{\rho_i^2}{2\sigma_0^2} - \log\frac{\sigma_0}{\sigma_1} - \log\frac{p_1}{p_0} \right] + \beta R(\mathbf{u}) \right\}. \tag{9.19}$$

where p_0 and p_1 are the prior probability of forged and non-forged, respectively, and β is the edge-penalty parameter indicating how strong the interaction between pixels, and the regularization term $R(\mathbf{u}) = \sum_{i=1}^{M\times N}\sum_{j\in\mathscr{N}_i}|u_j-u_i|$, with $\mathscr{N}_i$ the set of 4-connected neighbors of pixel i, is the sum of all class transitions over all 4-connected cliques of the image. By resorting to the convex-optimization algorithm proposed in [80], the $\hat{\boldsymbol{u}}$ that gives the maximal probability can be obtained. This method incorporates the prior information and spatial dependencies between pixels and therefore produces a more consistent and smooth binary map.

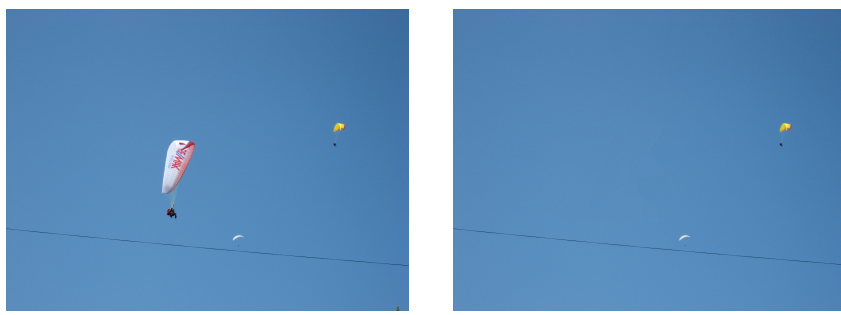
As can be observed in the detection results presented in [46] as well as the authentication map in Fig. 9.9, the falsely identified areas are largely located along the boundary of the forged area. The reason is that, when the detection block falls near the boundary between two different regions (i.e., forged and non-forged regions), the decision statistic ρ is a weighted average of two different contributions and more likely to exceed the decision threshold. As a result, missing detection occurs along the boundary between forged and non-forged regions.

Chierchia *et al.* [81] alleviated this problem by first segmenting the image under investigation and then calculating the decision statistic on the intersection of the detection block and the segmented objects. However, this method heavily depends on the performance of image segmentation, which itself is an ill-posed problem.

In view of this, Chierchia *et al.* [82] proposed an algorithm based on the guided filtering [83] to avoid the unreliable image segmentation. The basic idea is to post-process the calculated correlation map $\boldsymbol{\rho}$ by resorting to a pilot image, which can be a combination of the color bands of the original image or its denoised version, or any suitable field of features extracted from images [82]. The pilot image bears some valuable information, such as geometrical structures, of the image content and can be viewed as the soft-segmented version of the original image. By incorporating the structure information from the pilot image, the guided filtering is aware of the object boundaries and thus facilitates the decision-making process near boundaries. However, both the segmentation based method [81] and the soft segmentation based method [82] will fail when objects in the original scene are hidden by placing a homogeneous background on them, e.g., an airplane is covered by a patch of blue sky, or objects are removed by image inpainting or texture synthesis. One such example is shown in Fig. 9.10, where the paraglider and the pilot are removed by inpainting without leaving any visible traces. The segmentation based methods are unable to detect the forgeries in this case because no structure information is available in the forged regions.

In view of the limitations of the segmentation based detection algorithms, Lin and Li [84] proposed an algorithm to alleviate the missing detection problem along the boundary between forged and non-forged regions. They first applied the CFAR method [46] with two thresholds α and β to obtain an initial detection result Ω. Although the CFAR method suffers from the missing detection problem, it provides an indication that the nearby forged regions of the boundary of Ω may have been missed. So they modeled how the distribution of decision statistics changes as the detection block moves across the boundary of Ω and adjusted the threshold t in Eq. (9.16) for each pixel q:

$$\int_{t'(q)}^{\infty} p_{\Omega}(x,q)\mathrm{d}x = \alpha, \tag{9.20}$$



(a) (b)

Figure 9.10: Object removal by image inpainting. The segmentation based methods are unable to detect the forgeries in this case. (a) Authentic image. (b) Forged image.

where $t'(q)$ is the adjusted threshold for pixel q, and $p_\Omega(x,q)$ is the distribution of decision statistics depending on the initially detected forged region Ω and how far pixel q is from Ω. With the adjusted threshold, a pixel q is labeled as forged if the decision statistic $\rho_q < t'(q)$. Similar to [46], to reduce the false positives, a forged pixel will be labeled as non-forged if $\int_{-\infty}^{t} p(x|H_1)\mathrm{d}x > \beta$. The algorithm does not require to segment the image but makes use of the detection result of the CFAR method to identify the forged pixels that have been missed.

9.6 Summary and Outlook

We have introduced different intrinsic device fingerprints and their applications in image provenance inference. Although with varying levels of accuracy, the device fingerprints arising from optical aberration, CFA interpolation, camera response function and in-device image compression are effective in differentiating devices of different brands or models. Although they can not uniquely identify the source device of an image, they do provide useful information about the image provenance and are effective at narrowing down the image source to a smaller set of possible devices. More than half of the chapter was spent on Sensor Pattern Noise, which is the only fingerprint that distinguish devices of the same model. Because of its merits, such as the uniqueness to individual device and the robustness against common image operations, it has attracted much attentions from researches and been successfully used for source device identification, device linking, source-oriented image clustering and image forgery detection. In spite of the effectiveness of SPN, it is by nature a very weak signal and may have been contaminated by image content and other interferences. Its successful application requires jointly processing a large number of pixels, which results in very high dimensionality of SPN. This may bring huge difficulties in practice, e.g., in large-scale source-oriented image clustering based on SPN, so it is essential to conduct research on the compact representation of SPN for fast search and clustering.

With the development of Internet and the rise of big data, the amount of data generated in different knowledge areas has explosively increased. Big data opens new opportunities in identifying potential evidence from massive information, but the large scale size of digital images also presents new challenges for image provenance inference. Taking the source-oriented image clustering for example, the large scale size of image database impose a heavy burden on computation. One feasible solution would be combining the information from different fingerprints of an image to reduce the computational cost. For example, if we make use of the information from optical aberration, CFA interpolation, or even metadata of an image, and first partition the images of "similar" provenance into the same group, the computational complexity of the source-oriented image clustering algorithm in [75] can be significantly reduced.

Another line of research is the information integration of different fingerprints. Although attempts have been made to jointly use different fingerprints for image provenance inference, such as the work in [11], a universal and effective tool that allows the forensic investigators to synergically exploit these fingerprints and finally

reach a decision based on their output, is still lacking. This is by no means an easy task, because it requires to carefully investigate the performance of each fingerprint under different application scenarios and employ a decision fusion engine to reach a comprehensive and informative conclusion. Further research will be required to integrate the information from different device fingerprints.

References

[1] F. Mintzer, G. W. Braudaway, and M. M. Yeung. Effective and ineffective digital watermarks. In *Proceedings of International Conference on Image Processing*, volume 3, pages 9–12. IEEE, 1997.

[2] M. M. Yeung and F. C. Mintzer. Invisible watermarking for image verification. *Journal of Electronic Imaging*, 7(3):578–591, 1998.

[3] C.-Y. Lin and S.-F. Chang. Semifragile watermarking for authenticating jpeg visual content. In *Electronic Imaging*, pages 140–151. International Society for Optics and Photonics, 2000.

[4] P. Wong and N. Memon. Secret and public key image watermarking schemes for image authentication and ownership verification. *IEEE transactions on image processing*, 10(10):1593–1601, 2001.

[5] C.-T. Li and F.-M. Yang. One-dimensional neighborhood forming strategy for fragile watermarking. *Journal of Electronic Imaging*, 12(2):284–291, 2003.

[6] C.-T. Li. Digital fragile watermarking scheme for authentication of jpeg images. *IEE proceedings-vision, image and signal processing*, 151(6):460–466, 2004.

[7] C.-T. Li and Y. Yuan. Digital watermarking scheme exploiting nondeterministic dependence for image authentication. *Optical Engineering*, 45(12):127001–127001, 2006.

[8] X. Zhu, A. T. Ho, and P. Marziliano. A new semi-fragile image watermarking with robust tampering restoration using irregular sampling. *Signal Processing: Image Communication*, 22(5):515–528, 2007.

[9] X. Zhao, A. T. Ho, H. Treharne, V. Pankajakshan, C. Culnane, and W. Jiang. A novel semi-fragile image watermarking, authentication and self-restoration technique using the slant transform. In *International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP)*, volume 1, pages 283–286, 2007.

[10] P. Alvarez. Using extended file information (EXIF) file headers in digital evidence analysis. *International Journal of Digital Evidence*, 2(3):1–5, 2004.

[11] E. Kee, M. K. Johnson, and H. Farid. Digital image authentication from jpeg headers. *IEEE Transactions on Information Forensics and Security*, 6(3):1066–1075, 2011.

[12] Phil, H. Exiftool, 2016. http://owl.phy.queensu.ca/~phil/exiftool/.

[13] V. N. Mahajan. *Optical imaging and aberrations: part I: ray geometrical optics*. Bellingham: SPIE-The International Society for Optical Engineering, 1998.

[14] M. K. Johnson and H. Farid. Exposing digital forgeries through chromatic aberration. In *Proceedings of the 8th workshop on Multimedia and security*, pages 48–55. ACM, 2006.

[15] L. T. Van, S. Emmanuel, and M. S. Kankanhalli. Identifying source cell phone using chromatic aberration. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 883–886. IEEE, 2007.

[16] K. San Choi, E. Y. Lam, and K. K. Wong. Source camera identification using footprints from lens aberration. In *Electronic Imaging*, pages 60690J–60690J, 2006.

[17] K. San Choi, E. Y. Lam, and K. K. Wong. Automatic source camera identification using the intrinsic lens radial distortion. *Optics express*, 14(24):11551–11565, 2006.

[18] F. Devernay and O. D. Faugeras. Automatic calibration and removal of distortion from scenes of structured environments. In *SPIE's 1995 International Symposium on Optical Science, Engineering, and Instrumentation*, pages 62–72. International Society for Optics and Photonics, 1995.

[19] A. Fischer and T. Gloe. Forensic analysis of interdependencies between vignetting and radial lens distortion. In *IS&T/SPIE Electronic Imaging*, pages 86650D–86650D. International Society for Optics and Photonics, 2013.

[20] J. Yu, S. Craver, and E. Li. Toward the identification of dslr lenses by chromatic aberration. In *IS&T/SPIE Electronic Imaging*, pages 788010–788010. International Society for Optics and Photonics, 2011.

[21] A. C. Popescu and H. Farid. Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on Signal Processing*, 53(2):758–767, 2005.

[22] S. Bayram, H. Sencar, N. Memon, and I. Avcibas. Source camera identification based on cfa interpolation. In *IEEE International Conference Image Processing (ICIP)*, volume 3, pages III–69, 2005.

[23] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[24] P. Pudil, FJ Ferri, J. Novovicova, and J. Kittler. Floating search methods for feature selection with nonmonotonic criterion functions. In *Proceedings of International Conference on Pattern Recognition*. Citeseer, 1994.

[25] S. Bayram, H. T. Sencar, N. Memon, and I. Avcibas. Improvements on source camera-model identification based on cfa interpolation. *Proc. of WG*, 11:24–27, 2006.

[26] Y. Long and Y. Huang. Image based source camera identification using demosaicking. In *Proceedings of IEEE Workshop on Multimedia Signal Processing*, 2006.

[27] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[28] A. Swaminathan, M. Wu, and KJ R. Liu. Non-intrusive forensic analysis of visual sensors using output images. In *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 5, pages V–V. IEEE, 2006.

[29] A. Swaminathan, M. Wu, and K.J. Liu. Nonintrusive component forensics of visual sensors using output images. *IEEE Transactions on Information Forensics and Security*, 2(1):91–106, 2007.

[30] H. Cao and A. C. Kot. Accurate detection of demosaicing regularity for digital image forensics. *IEEE Transactions on Information Forensics and Security*, 4(4):899–910, 2009.

[31] H. Cao and A. C. Kot. Mobile camera identification using demosaicing features. In *Proceedings of IEEE International Symposium on Circuits and Systems*, pages 1683–1686. IEEE, 2010.

[32] S. Lin, J. Gu, S. Yamazaki, and H.-Y. Shum. Radiometric calibration from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–938. IEEE, 2004.

[33] Z. Lin, Wang R., Tang X., and Shum H.-Y. Detecting doctored images using camera response normality and consistency. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 1087–1092, June 2005.

[34] M. D. Grossberg and S. K. Nayar. What is the space of camera response functions? In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–602. IEEE, 2003.

[35] Y.-F. Hsu and S.-F. Chang. Detecting image splicing using geometry invariants and camera characteristics consistency. In *IEEE International Conference on Multimedia and Expo*, pages 549–552, 2006.

[36] Y.-F. Hsu and S.-F. Chang. Image splicing detection using camera response function consistency and automatic segmentation. In *2007 IEEE International Conference on Multimedia and Expo*, pages 28–31. IEEE, 2007.

[37] Y.-F. Hsu and S.-F. Chang. Camera response functions for image forensics: an automatic algorithm for splicing detection. *IEEE Transactions on Information Forensics and Security*, 5(4):816–825, 2010.

[38] T.-T. Ng, S.-F. Chang, and M.-P. Tsui. Using geometry invariants for camera response function estimation. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[39] T. T. Ng and M. P. Tsui. Camera response function signature for digital forensics - part i: Theory and data selection. In *2009 First IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 156–160, Dec 2009.

[40] T.-T. Ng. Camera response function signature for digital forensics-part ii: Signature extraction. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 161–165. IEEE, 2009.

[41] H. Farid. Digital image ballistics from JPEG quantization. *Department of Computer Science, Dartmouth College, Tech. Rep. TR2006-583*, 2006.

[42] H. Farid. Digital image ballistics from JPEG quantization: A followup study. *Department of Computer Science, Dartmouth College, Tech. Rep. TR2008-638*, 2008.

[43] M. J. Sorell. Digital camera source identification through JPEG quantisation. *Multimedia forensics and security*, pages 291–313, 2008.

[44] E. Kee and H. Farid. Digital image authentication from thumbnails. In *IS&T/SPIE Electronic Imaging*, pages 75410E–75410E. International Society for Optics and Photonics, 2010.

[45] J. Lukas, J. Fridrich, and M. Goljan. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2):205–214, 2006.

[46] M. Chen, J. Fridrich, M. Goljan, and J. Lukás. Determining image origin and integrity using sensor noise. *IEEE Transactions on Information Forensics and Security*, 3(1):74–90, 2008.

[47] M. Goljan. Digital camera identification from images estimating false acceptance probability. In *Digital Watermarking*, volume 5450, pages 454–468. 2009.

[48] X. Kang, Y. Li, Z. Qu, and J. Huang. Enhancing source camera identification performance with a camera reference phase sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 7(2):393–402, 2012.

[49] M. Goljan, J. Fridrich, and T. Filler. Large scale test of sensor fingerprint camera identification. In *IS&T/SPIE Electronic Imaging*, pages 72540I–72540I, 2009.

[50] M.K. Mhak, I. Kozintsev, and K. Ramchandran. Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 6, pages 3253–3256, Mar 1999.

[51] G. Chierchia, S. Parrilli, G. Poggi, C. Sansone, and L. Verdoliva. On the influence of denoising in prnu based forgery detection. In *ACM Workshop on Multimedia in Forensics, Security and Intelligence*, pages 117–122, NY, USA, 2010.

[52] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, Aug 2007.

[53] G. Wu, X. Kang, and K. R. Liu. A context adaptive predictor of sensor pattern noise for camera source identification. In *IEEE International Conference on Image Processing (ICIP)*, pages 237–240, 2012.

[54] X. Kang, J. Chen, K. Lin, and P. Anjie. A context-adaptive SPN predictor for trustworthy source camera identification. *EURASIP Journal on Image and Video Processing*, 2014(1):1–11, 2014.

[55] C.-T. Li. Source camera identification using enhanced sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 5(2):280–287, 2010.

[56] X. Lin and C.-T. Li. Enhancing sensor pattern noise via filtering distortion removal. *IEEE Signal Processing Letters*, 23(3):381–385, March 2016.

[57] S. McCloskey. Confidence weighting for sensor fingerprinting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–6, June 2008.

[58] L.-H. Chan, N.-F. Law, and W.-C. Siu. A confidence map and pixel-based weighted correlation for prnu-based camera identification. *Digital Investigation*, 10(3):215–225, 2013.

[59] Y. Hu, B. Yu, and C. Jian. Source camera identification using large components of sensor pattern noise. In *International Conference on Computer Science and its Applications*, pages 291–294, 2009.

[60] R. Li, C.-T. Li, and Y. Guan. A reference estimator based on composite sensor pattern noise for source device identification. In *IS&T/SPIE Electronic Imaging*, pages 90280O–90280O. International Society for Optics and Photonics, 2014.

[61] A. Lawgaly, F. Khelifi, and A. Bouridane. Weighted averaging-based sensor pattern noise estimation for source camera identification. In *IEEE International Conference Image Processing (ICIP)*, pages 5357–5361, Oct 2014.

[62] C.-T. Li and Y. Li. Color-decoupled photo response non-uniformity for digital image forensics. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(2):260–271, 2012.

[63] M. Goljan and J. Fridrich. Sensor-fingerprint based identification of images corrected for lens distortion. In *IS&T/SPIE Electronic Imaging*, pages 83030H–83030H. International Society for Optics and Photonics, 2012.

[64] M. Goljan and J. Fridrich. Estimation of lens distortion correction from single images. In *IS&T/SPIE Electronic Imaging*, pages 90280N–90280N. International Society for Optics and Photonics, 2014.

[65] T. Gloe, S. Pfennig, and M. Kirchner. Unexpected artefacts in PRNU-based camera identification: A 'Dresden Image Database' Case-Study. In *ACM Workshop on Multimedia and Security*, pages 109–114, 2012.

[66] X. Lin and C.-T. Li. Preprocessing Reference Sensor Pattern Noise via Spectrum Equalization. *IEEE Transactions Information Forensics and Security*, 11(1):126–140, 2016.

[67] R. Satta and P. Stirparo. On the usage of sensor pattern noise for picture-to-identity linking through social network accounts. In *Proceedings of International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 3, pages 5–11. IEEE, 2014.

[68] M. Goljan, M. Chen, and J. Fridrich. Identifying common source digital camera from image pairs. In *Proceedings of International Conference on Image Processing*, pages 125–128, 2007.

[69] G. J. Bloy. Blind camera fingerprinting and image clustering. *IEEE Transactions Pattern Analysis Machine Intelligence*, 30(3):532–534, 2007.

[70] C.-T. Li. Unsupervised classification of digital images using enhanced sensor pattern noise. In *IEEE International Symposium on Circuits and Systems*, pages 3429–3432, May 2010.

[71] B.-B. Liu, H.-K. Lee, Y. Hu, and C.-H. Choi. On classification of source cameras: A graph based approach. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–5, Dec 2010.

[72] S. X. Yu and J. Shi. Multiclass spectral clustering. In *the 9th IEEE International Conference Computer Vision*, pages 313–319, 2003.

[73] R. Caldelli, I. Amerini, F. Picchioni, and M. Innocenti. Fast image clustering of unknown source images. In *IEEE International Workshop on Information Forensics and Security*, pages 1–5, Dec 2010.

[74] L. J. G. Villalba, A. L. S. Orozco, and J. R. Corripio. Smartphone image clustering. *Expert System with Applications*, 42(4):1927–1940, 2015.

[75] X. Lin and C.-T. Li. Large-scale image clustering based on device fingerprints. *Accepted for publication in IEEE Transactions on Information Forensics and Security*, 2016.

[76] S. M. Van Dongen. *Graph clustering by flow simulation*. PhD thesis, University of Utrecht, Netherlands, 2000.

[77] T. Gloe and R. Bhme. The Dresden Image Database for Benchmarking Digital Image Forensics. *Journal of Digital Forensic Practice*, 3(2-4):150–159, 2010.

[78] J. Lukáš, J. Fridrich, and M. Goljan. Detecting digital image forgeries using sensor pattern noise. In *SPIE*, pages 362–372, 2006.

[79] G. Chierchia, G. Poggi, C. Sansone, and L. Verdoliva. A Bayesian-MRF approach for PRNU-based image forgery detection. *IEEE Transactions on Information Forensics and Security*, 9(4):554–567, 2014.

[80] P. L. Combettes and J.-C. Pesquet. Primal-dual splitting algorithm for solving inclusions with mixtures of composite, lipschitzian, and parallel-sum type monotone operators. *Set-Valued and Variational Analysis*, 20(2):307–330, 2012.

[81] G. Chierchia, S. Parrilli, G. Poggi, L. Verdoliva, and C. Sansone. Prnu-based detection of small-size image forgeries. In *IEEE International Conference on Digital Signal Processing*, pages 1–6, July 2011.

[82] G. Chierchia, D. Cozzolino, G. Poggi, C. Sansone, and L. Verdoliva. Guided filtering for prnu-based localization of small-size image forgeries. In *IEEE International Conference Acoustics, Speech, Signal Processing (ICASSP)*, pages 6231–6235, 2014.

[83] K. He, J. Sun, and X. Tang. Guided image filtering. In *European Conference Computer Vision*, pages 1–14. Springer, 2010.

[84] X. Lin and C.-T. Li. Refining PRNU-based detection of image forgeries. In *Proceedings of Digital Media Industry Academic Forum (DMIAF)*, pages 222–226, July 2016.