JOURNAL OF FORENSIC SCIENCES

J Forensic Sci, March 2012, Vol. 57, No. 2
doi: 10.1111/j.1556-4029.2012.02057.x
Available online at: onlinelibrary.wiley.com

TECHNICAL NOTE

DIGITAL AND MULTIMEDIA SCIENCES

Wiger van Houten,[1,2] *M.Sc. and Zeno Geradts,*[1] *Ph.D.*

Using Anisotropic Diffusion for Efficient Extraction of Sensor Noise in Camera Identification

ABSTRACT: Each digital camera has an intrinsic fingerprint that is unique to each camera. This device fingerprint can be extracted from an image and can be compared with a reference device fingerprint to determine the device origin. The complexity of the filters proposed to accomplish this is increasing. In this note, we use a relatively simple algorithm to extract the sensor noise from images. It has the advantages of being easy to implement and parallelize, and working faster than the wavelet filter that is common for this application. In addition, we compare the performance with a simple median filter and assess whether a previously proposed fingerprint enhancement technique improves results. Experiments are performed on approximately 7500 images originating from 69 cameras, and the results are compared with this often used wavelet filter. Despite the simplicity of the proposed method, the performance exceeds the common wavelet filter and reduces the time needed for the extraction.

KEYWORDS: forensic science, camera identification, digital forensics, photo-response nonuniformity, source verification, sensor forensics

Digital photography was adopted in a short time to replace its analog counterpart in the last two decades. Along with this transition came the easy and widespread usage of the Internet, making it possible to share and copy these images without leaving behind any traces. In some situations, however, the question of the image source can be of paramount importance in a forensic context.

An obvious example is whether an image was downloaded or actually produced by a suspect in a child pornography case. Another possibility is to test testimonies and confessions, for example, whether a suspect was near a crime scene if photographs or videos were found on his mobile phone or camera. As a final example, the question of the origin of a stolen camera found at a suspect's residence, for example in a violent robbery, may be solved if the owner has some reference material available (e.g., on his computer).

There has been much effort in the forensics community to solve the question of camera identification based on the images a camera produces. Roughly, there are two approaches. The first is mostly a camera classification scheme based on statistical information from the images. This approach (e.g., see [1] and the references therein) uses Support Vector Machines (SVMs), to classify the images based on a large number of features. Examples of these features are Binary Similarity Measures in which the tell-tale patterns from the bitplanes are used as a characteristic of a camera brand/model, Image Quality Measures in which features such as image sharpness and other visual differences are used, Higher-Order Wavelet Statistics in which characteristics about the noise are exploited, demosaicking artifacts in which the Color Filter Array interpolation algorithm is used as a distinctive characteristic, and more (1). Although the performance can be quite good, it has the disadvantage that the SVM needs to be trained properly, which can take quite a lot of time as feature extraction for high-resolution images is computationally expensive, especially when the number of features is large. The potential advantage is that spatially transformed images may still be identified (1)—if the SVM is trained properly. In (2), an alternative approach is presented. As image sensors are monochrome devices, they cannot differentiate between different colors. To produce a color output, a Color Filter Array is placed on the top of the sensor. With this addition, each pixel absorbs light with a wavelength range corresponding to either the red, green, or blue colors (in the common Bayer filter array). To produce a full-color image, an interpolation step is needed. Different camera brands and models often use different interpolation algorithms to accomplish this. In this way, it is possible to perform camera classification by estimating the interpolation algorithm that has been used inside the camera. Likewise, in (3), singular value decomposition is used to detect the linear dependency in rows/columns owing to the interpolation artifacts.

The second approach does not rely on these statistically significant deviations, but instead uses characteristics that individualize each sensor. When digital cameras started to gain popularity, CCD and CMOS sensors contained defects (dead or hot pixels). These defects in turn created a unique pattern that could help linking photographs and their sensors (4). Owing to the possible absence of these defects in modern sensors, the forensics community started looking at the individual deviation of each pixel (5). This is the

[1]Digital Technology and Biometrics Department, Netherlands Forensic Institute, Laan van Ypenburg 6, 2497 GB, The Hague, The Netherlands.
[2]Current address: Northern Criminal Investigation Unit, Department of Digital Evidence, Schweitzerlaan 1, 9728 NP Groningen, The Netherlands.

Received 4 May 2010; and in revised form 7 Jan. 2011; accepted 9 Jan. 2011.

photo-response nonuniformity (PRNU) and is based on the characteristic fingerprint the camera unintentionally leaves behind in each image it produces. The method of operation relies on the fact that each pixel has a slightly different response to the same amount of light, creating a characteristic pattern of deviations. The presence of this fingerprint depends among others on the intensity, as it is multiplicative. Images that contain high-frequency textures or large dark areas may be difficult to identify. This is due to information loss occurring most often in JPEG (Joint Photographic Experts Group, a popular image compression format) compressed images in high-frequency textures and the absence of the pattern in dark areas. In general, however, this pattern is quite robust against compression, and under certain conditions even works for videos from YouTube (6). However, spatial transformations desynchronize the PRNU pattern (e.g., cropping, rotation, or scaling) and make identification harder or near impossible. When images are cropped and/or scaled, identification is still possible as was shown in (7).

The techniques used to denoise the images are becoming more and more advanced (see, e.g., [8]), and with this, the computational complexity is increasing. Instead of using the most advanced method to denoise the images, we found an efficient multiscale algorithm (9) that is very easy to implement.

This study is organized as follows. In the next section, we will expand on a few general things about the method of identification, and briefly explain the algorithm used to extract the PRNU pattern from the images. Subsequently, after explaining the experimental conditions in the third section, we will use this algorithm to measure the performance in the fourth section. After a discussion in the fifth section, the study is concluded in the last section.

Methods

The origin of the PRNU lies in construction and device nonidealities. Specifically, this means that when all pixels are equally illuminated, the output from these pixels will be slightly different. These variations are owing to nonuniform sizes of the active area of the pixels or nonuniform potential wells resulting in a different spectral response. Hence, some pixels may collect more photons as a result of a larger size of the photo transistor or absorb more long-wavelength photons as its potential well may be deeper.

There are two opposing trends making and breaking this scheme: on the one hand, owing to improving manufacturing standards, the presence of the PRNU may decrease as technology advances. On the other hand, the resolution increases as well, resulting in smaller pixels and relatively larger deviations.

As put forth in the Introduction, the camera identification scheme relies on the extraction of the characteristic digital fingerprint from photographs. Although this pattern is often imperceptible from the image itself, it is possible to extract it from the image with advanced filters. To conclude that a certain image was made with a certain camera, we need to compare the pattern from the questioned image with a reference pattern of the suspected camera. In general, extracting the pattern from an image is easiest when the image contains no textures or edges. Furthermore, as the PRNU is multiplicative (its effect increases when the illumination increases), it is preferred to use images that are reasonably illuminated (not saturated). These kinds of images are called flatfield or reference images and can be made by photographing out-of-focus bright skies or flat surfaces such as desks. To calculate the reference pattern of a camera, a large number (we used 50) of these images are made, after which the patterns from each single image are extracted and averaged. By averaging these individual patterns, temporary fluctuations (e.g., photon shot noise) are averaged out, thus obtaining the reference pattern. In addition to flatfield or reference images, there are regular images (typically including actual scene content) to which we refer to as "natural images." Now, the fingerprint from a questioned image of unknown origin is compared with the reference pattern, often simply by calculating Pearson's correlation. A high correlation suggests a (linear) relationship between these two seemingly random patterns. This method works reliably even when large numbers of images and cameras are used. In (10), the results of a large-scale experiment (over one million images, spanning almost 7000 different cameras) are reported with very low false rejection rates (FRR) and false acceptance rates (FAR).

In actual casework, we prefer to use a large number (at least 10) of reference cameras of the same make/model to exclude the possibility of matching based on class-specific characteristics. Specifically, it is possible that class characteristic patterns (e.g., from CFA interpolation, JPEG compression artifacts) may be present in the extracted pattern (11). These characteristic patterns may be specific to a class of cameras, for example, the brand and/or model. Hence, in these cases, an elevated correlation value does not signify identification, but merely shows a different relationship between the two patterns exists. Using multiple reference cameras, we can exclude the possibility of matching based on just these characteristics. These considerations are especially important with low-quality (high compression) images and/or videos.

Algorithm

We will now briefly explain the algorithm to extract the PRNU pattern from images. For a more in-depth explanation, we refer to the original publications (9,12). As put forth in the previous paragraph, extracting the sensor noise P from the image is carried out by simply subtracting a denoised (filtered) version $F(I)$ from the original image I:

$$P = I - F(I) \tag{1}$$

An image may consist of various homogeneous and inhomogeneous areas. Intuitively speaking, there may be continuous (smooth) areas of approximately the same intensity and discontinuous areas (textures, edges) with variable intensities. In this framework, we can understand the anisotropic diffusion algorithm (9) by first considering the continuity equation, in which we explicitly assume that the intensity $I(x, y, t)$ is a conserved quantity:

$$\frac{\partial I(x,y,t)}{\partial t} = -\nabla \cdot J(x,y,t) \tag{2}$$

in which $J(x, y, t)$ denotes the flux of the image intensity, and the whole right-hand side of the equation denotes the divergence of the flux. The t-variable in $I(x, y, t)$ signifies the multiscale approach; each t signifies a different scale. Hence, this equation states that the image intensity is simply redistributed in the image and the rate at which this happens equals the negative divergence of the flux. In other words, there is a redistribution of the pixel values in close proximity of each other.

The flux, in turn, can be described by:

$$J = -c(x,y,t)\nabla I(x,y,t) \tag{3}$$

This states that the high image intensity values "flow" to lower intensity values, depending on the gradient of the image, and the diffusion coefficient c. Combining these two equations, we find the anisotropic diffusion equation (9):

$$\frac{\partial I(x,y,t)}{\partial t} = \nabla \cdot (c\nabla I) = c\nabla^2 I + \nabla c \cdot \nabla I = \frac{\partial}{\partial x}(c\frac{\partial I}{\partial x}) + \frac{\partial}{\partial y}(c\frac{\partial I}{\partial y}) \tag{4}$$

Depending on the (local) diffusion coefficient c, the image is denoised. If $c(x, y, t) = 1\ \forall(x, y)$, then we can see that

$$I(x,y,t) = I_0(x,y,t) * G(x,y,t), \tag{5}$$

$$G(x,y,t) = \frac{1}{2\pi t} e^{-(x^2+y^2)/2t} \tag{6}$$

is a valid solution; this denotes *isotropic* diffusion. However, isotropic diffusion results in the blurring of edges and textures which we want to avoid because this will lead to image residue in the PRNU pattern. Therefore, optimizing the denoising comes down to finding suitable diffusion coefficients. The authors (9) propose to use the image gradient as a parameter to control the diffusion. This gives rise to the anisotropic diffusion (a different diffusion parameter in each direction).

Perona and Malik (9) chose to use the four nearest neighbors, and this was later extended to the eight nearest neighbors in (12).

Using the difference quotients as an approximation of the derivatives (4), we find that (9)

$$\frac{\partial I(x,y,t)}{\partial t} = c_N \Delta I_N - c_S \Delta I_S + c_E \Delta I_E - c_W \Delta I_W \tag{7}$$

where Δ denotes nearest neighbors differences. For example, ΔI_N denotes the difference between the current pixel and the pixel above it (''North''), and c_N denotes the diffusion parameter for this direction. Intuitively, when the image I is smooth, the diffusion parameter should be close to 1: This approximates isotropic diffusion (''Gaussian blurring''). On the other hand, when the image contains textures the diffusion parameter will be smaller to prevent edge distortion at boundaries. After the intensity values have been redistributed, the image is adjusted to reflect the new intensities:

$$I^{t+1} = I^t + \lambda(c_N \Delta I_N - c_S \Delta I_S + c_E \Delta I_E - c_W \Delta I_W) \tag{8}$$

where λ is the integration constant ($0 \le \lambda \le 1/3$ for four neighbors). To obtain the denoised image at a certain scale t, the image at scale t-1 is denoised. The first scale can be obtained by denoising the original image (scale 0). Finally, a small λ gives a better approximation of the original equation (4), but we will need more iterations to denoise the image.

Adding the four diagonal neighbors results in

$$\begin{aligned} I^{t+1} = I^t + \lambda((&c_N \Delta I_N - c_S \Delta I_S + c_E \Delta I_E - c_W \Delta I_W) \\ &+ \frac{1}{2}(c_{NW} \Delta I_{NW} - c_{NE} \Delta I_{NE} + c_{SE} \Delta I_{SE} - c_{SW} \Delta I_{SW})) \end{aligned} \tag{9}$$

with $0 \le \lambda \le 1/7$ (11). Note the factor ½ owing to the larger distance to the diagonal pixel. Owing to this larger distance, this pixel should have less influence on the pixel we are considering. As we are applying the second derivative, this gives a factor of ½.

The ΔI_N term can be obtained from a simple convolution:

$$\Delta I_N = I * g, g = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{10}$$

and likewise for the other directions.

Finally, it is necessary to know where the diffusion needs to occur and where not; that is, we need to have an edge estimate to obtain the diffusion parameter. Perona and Malik (9) use the gradient of the image as the diffusion parameter. A small gradient occurs where the area is homogeneous, which is where we want the diffusion (hence a large diffusion parameter) to occur and vice versa. They propose two different diffusion functions based on the gradient:

$$\begin{aligned} c(x,y,t) &= \exp(-(|\Delta I|/K)^2 \\ c(x,y,t) &= \frac{1}{1 + (|\Delta I|/K)^2} \end{aligned} \tag{11}$$

The value for K is determined at each iteration. First, the gradient of the whole image at the previous scale is calculated (at the first scale, the original image is taken). After taking the absolute value of this gradient, the histogram is calculated, and the value below which 90% of the intensity values occur is denoted as K. After a set number of iterations, the denoised image $F(I)$ is obtained.

After the filtered image $F(I)$ has been calculated, the pattern is obtained by subtracting this filtered image from the original image (1). As rightfully noted in (11), artifacts in the pattern exist owing to (class) characteristic CFA interpolation and JPEG compression. These characteristic artifacts result in a similarity between two otherwise unrelated patterns. We follow (11) by suppressing these artifacts by subsequently subtracting the column and row averages from the obtained pattern.

Although it may look complicated, this algorithm is very easy to implement. In the Appendix, pseudo-code can be found. The convolutions are actually simple subtractions, and the other calculations are all carried out pointwise, which means that this algorithm is O(N). Finally, this approach makes it trivial to implement the algorithm in parallel to take advantage of multicore processing (e.g., each convolution executed in a separate thread).

Experimental Settings

The algorithm was implemented in Matlab 2009b (http://mathworks.com/), but no explicit multithreading was implemented. Owing to its simplicity and the use of standard functions, the amount of code is limited to approximately 50 lines. To compare the performance of the wavelet algorithm (5), we did the same for this algorithm. The Wavelab850 toolbox (13) was used for the latter algorithm. The improvements in (11) to reduce periodic artifacts, namely zero-meaning and wiener-filtering, were also implemented. In (5), it was suggested that nondyadic images could be processed by blocks. The advantage is that the memory needed for calculating the PRNU pattern remains within boundaries. On the other hand, processing nonoverlapping blocks results in boundary effects. Instead, we adjusted the code from Wavelab850 to allow for nonsquare nondyadic images. This significantly reduces the overhead from overlapping blocks. The downside is that it uses more memory than when (smaller) individual blocks are processed. For comparison, we used the Daubechies wavelet as well as the Coiflet wavelet and varied the denoising parameter σ = 1–4. It

would be advantageous to estimate the denoising parameter from the image. Indeed, when an image is corrupted with White Gaussian Noise, this can be done easily be transforming the image to the wavelet domain and subsequently calculate the median in the highest-level subband (14). However, this does not work reliably in normal (uncorrupted) images, and hence, we use a fixed sigma.

For the anisotropic diffusion algorithm, we chose the first diffusion function owing to its slightly better performance in our tests. The number of scales was set to three, and λ was fixed to 1/7. We tested four as well as eight local neighbors.

Finally, we tested whether the work presented in (15) to reduce the image residue in the PRNU pattern resulted in a better performance, for both the wavelet method and the presented method. This is accomplished by attenuating pixels that show a very strong deviation in the PRNU pattern, as these pixels are likely to originate from image content. The larger the magnitude of the pixel in the PRNU pattern, the more it is attenuated.

The experiments were performed on an Intel Xeon E5410 2.3 GHz with 4GB of RAM (Intel Corp., Santa Clara, CA). We used 69 cameras for the comparison of three different brands comprising seven different models and resolutions, as can be seen in Table 1. In this table, the number of natural images is also presented. The number of reference images used for the calculation of the reference patterns was 50 for each camera used.

For each camera, a number of natural images were made, ranging from 50 to 150 per camera. The images were made inside and outside the office, with a wide range of textures, details, and illuminations (saturation was frequently present). Images were captured in automatic setting, with digital zoom turned off.

To judge whether an image originates from a certain camera, the correlation coefficient is used. It is defined in the usual way:

$$\rho(n, R) = \frac{(n - \overline{n})(R - \overline{R})}{||n - \overline{n}||||R - \overline{R}||} \qquad (12)$$

where n is the PRNU pattern of the natural image, R is the reference pattern, and the bar above n and R denotes the average.

Results

A total of 7502 natural images were used to assess the performance of both methods. For each model and method, we experimentally determined the equal error rate (EER), defined as the point where the FAR and FRR are equal.

In Table 2, the EER of the wavelet method (Daubechies wavelet) are presented for σ = 1–4, for each camera. Using a Coiflet wavelet did not improve the result (on average, approximately 50% higher EERs). On the basis of these results, it was decided to use σ = 2 (Daubechies) in the following comparisons.

In Table 3, the results are presented for both the wavelet method and the proposed method (columns 1 and 2). For the Sony DSC-S930 (Sony Corp., Tokyo, Japan), we see a dramatic rise of the EER. Upon inspection, we see that the distribution of the correlation values for the mismatching pairs (the correlation between an image and a reference pattern from different cameras) has dramatically shifted to higher values. This means that the patterns still contain class-specific characteristics of the model/brand camera.

Changing the number of scales/iterations did not significantly improve the results. When random images were inspected, we realized that most surfaces are locally approximately smooth or even uniform in their intensity, owing to the Lambertian reflectance property. Specifically, this means that a surface has the same brightness irrespective of the angle of view (isotropic luminance). Therefore, it was suggested that applying a median filter would do the initial denoising, after which the anisotropic diffusion was again applied. This median filter should help to reduce the amount of (small) impulse noise. It works by sliding a 3×3 window over each pixel and substituting the center pixel value by the median of the sorted nine values inside the window. Note that this median filter was only applied to the natural images.

Results for this combined approach are presented in the last column of Table 3, in which we see the improved sensitivity of the method. As the median filter only performs local operations, it only results in a time penalty of approximately 10% with respect to the proposed algorithm. Interestingly, using only four local neighbors instead of eight resulted in the best performance.

When this median filter was applied for the wavelet method results did not improve, but gave on average a slight performance penalty. Finally, we also attempted to use the median filter for the PRNU extraction. The performance was much worse than the other two filters, as can be seen in Table 4.

TABLE 1—*Overview of the number of cameras used and their specifications.*

Brand	Model	Resolution (MP)	Cameras	Images
Samsung	Digimax S500	2560 × 1920 (5)	10	998
Samsung	Digimax L70	3072 × 2304 (7)	10	1060
Canon	PowerShot A430	2272 × 1704 (4)	10	1050
Canon	PowerShot A630	3264 × 2448 (8)	10	500
Sony	CyberShot DSC-S500	2816 × 2112 (6)	9	900
Sony	CyberShot DSC-S800	3264 × 2448 (8)	10	1494
Sony	CyberShot DSC-S930	3648 × 2736 (10)	10	1500

TABLE 2—*Performance measures for the wavelet method.*

Brand	Model	σ = 1	σ = 2	σ = 3	σ = 4
Samsung	Digimax S500	0.50	0.50	0.40	0.57
Samsung	Digimax L70	0.00	0.00	0.00	0.00
Canon	PowerShot A430	1.14	0.86	0.76	0.57
Canon	PowerShot A630	0.00	0.00	0.00	0.00
Sony	CyberShot DSC-S500	0.00	0.33	1.22	1.89
Sony	CyberShot DSC-S800	0.47	0.47	0.40	0.40
Sony	CyberShot DSC-S930	10.8	4.93	4.73	4.23
	Weighted Average	2.48	1.30	1.33	1.31

Equal error rates (percentage) for the wavelet method (Daubechies wavelet) for different denoising parameters. From this, it was concluded that σ = 2 gave the best performance (lowest number of errors).

TABLE 3—*Performance measures for both methods.*

Brand	Model	EER^{w} (%)	EER^{ad} (%)	$EER^{ad'}$ (%)
Samsung	Digimax S500	0.50	0.20	0.20
Samsung	Digimax L70	0.00	0.00	0.00
Canon	PowerShot A430	0.86	0.42	0.09
Canon	PowerShot A630	0.00	0.00	0.00
Sony	CyberShot DSC-S500	0.33	0.00	0.00
Sony	CyberShot DSC-S800	0.47	0.50	0.47
Sony	CyberShot DSC-S930	4.93	11.0	0.60
	Weighted Average	1.31	2.38	0.25

EER, equal error rate.

EER^{w} denotes the equal error rate for the wavelet method, and EER^{ad} denotes the equal error rate for the anisotropic diffusion method. Finally, $EER^{ad'}$ denotes the equal error rate for the adjusted method, that is, the anisotropic diffusion method followed after the initial median filter (see the text).

TABLE 4—*Equal error rate (EER) for the median filter with and without reducing the image residue (12).*

Brand	Model	EER^{m} (%)	$EER^{m'}$ (%)
Samsung	Digimax S500	12.3	0.86
Samsung	Digimax L70	41.6	35.4
Canon	PowerShot A430	12.0	0
Canon	PowerShot A630	50.9	0.89
Sony	CyberShot DSC-S500	38.0	0.09
Sony	CyberShot DSC-S800	41.0	1.50
Sony	CyberShot DSC-S930	35.6	1.13
	Weighted Average	32.4	5.71

EER^{m} denotes the equal error rate for extracting the photo-response nonuniformity with the median filter, and $EER^{m'}$ denotes the equal error rate for when the image residue reduction is applied to the median filter.

Reducing the image residue as presented in (15) only improved the results very slightly for both the wavelet method and the proposed method. As noted in (15), this is expected, as the performance gain is mainly beneficial for low-resolution images. This shows that both methods are already sufficiently capable of discriminating between image features and noise. Interestingly, when the method was applied to the median filter, results improved dramatically, as seen in Table 4.

As a final assessment of the discriminative ability of the method, we also calculated the correlation between the PRNU patterns from each natural image and the reference PRNU pattern from each of the 69 cameras. To circumvent the problem of differing resolutions from the cameras and to speed up the extraction, we limited the pattern size to 1536×1536. In this way, $7502.68 \approx 5.1 \times 10^5$ mismatching correlations and 7502 matching correlations were calculated. This resulted in an EER of 2.81% for the wavelet method and an EER of 0.50% for the proposed method (including the median filter). The results from this experiment are summarized in Fig. 1. In this figure, we present the detection error trade-off (DET) curve, a plot (more common in biometrics) representing the false-positive rate versus the false-negative rate on log–log scales. This allows a better distinguishing between the relative errors than the common receiver operating characteristic curve. From this figure, we conclude that the proposed method is better suited for camera identification for all but the highest FARs. Specifically, at a FAR of 10^{-3}, the FRR for the proposed method is 1.4×10^{-2}, while the FRR is 0.27 for the wavelet method. At a FAR of 10^{-2}, the FRRs are 3.33×10^{-3} and 0.135, respectively.

A possible reason for the better performance of the proposed method is that the denoising only takes place in the immediate vicinity of each pixel. In contrast, for the wavelet method, the denoising is carried out in the wavelet domain in four levels. Especially in the larger/coarser scales, the denoising may quickly be too abrupt and/or cover a region that is too large.

Discussion

We inspected 40 images (≈0.5%) which were responsible for the lowest correlations in the match distribution, for both methods. More than 50% of these images were problematic for both methods. Upon inspection, we found that there were different sources of these problems: highly detailed textures (sand, gravel, leaves, twigs of trees, and fabrics), large dark or saturated areas (in general, images with large contrasts such as trees in snow), and elevated noise (higher ISO values).

In the histogram, it was observed that the distribution of the correlation values for nonmatching pairs (between the PRNU pattern from a natural image and a nonmatching reference pattern) is much narrower for the proposed method. On the other hand, the matching distributions are comparable, with the distribution for the proposed method skewed to the left.

It was suggested in (5) that the correlation values for the mismatching pairs for the wavelet method could be described be a generalized Gaussian distribution. However, we found that the logistic distribution better described the distribution of the proposed method because of its slightly wider tails:

$$f(x|\mu,\omega) = \frac{\exp(-(x-\mu)/\omega)}{\omega(1+\exp(-(x-\mu)/\omega))^{\omega}} \tag{13}$$

where f denotes the probability density function. In this formula, μ denotes the location and ω denotes the scale of the density

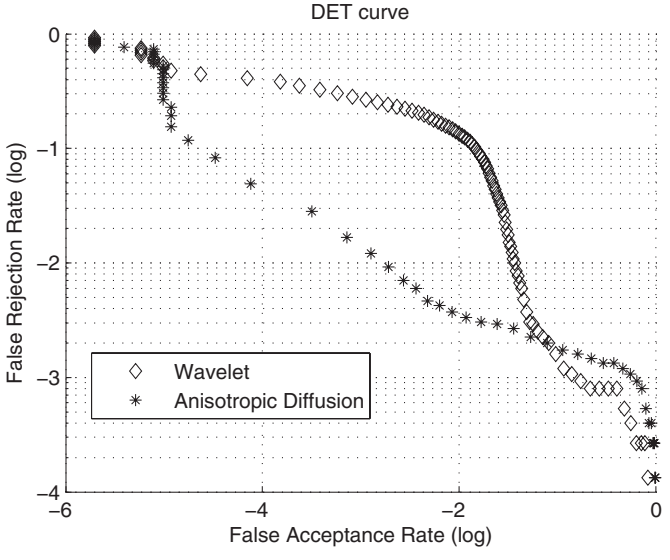

FIG. 1—*Detection error trade-off (DET) curve for the comparison of all images with all reference patterns for both methods. We see the improved performance (lower false acceptance and rejection rates) for all but the highest false acceptance rates.*

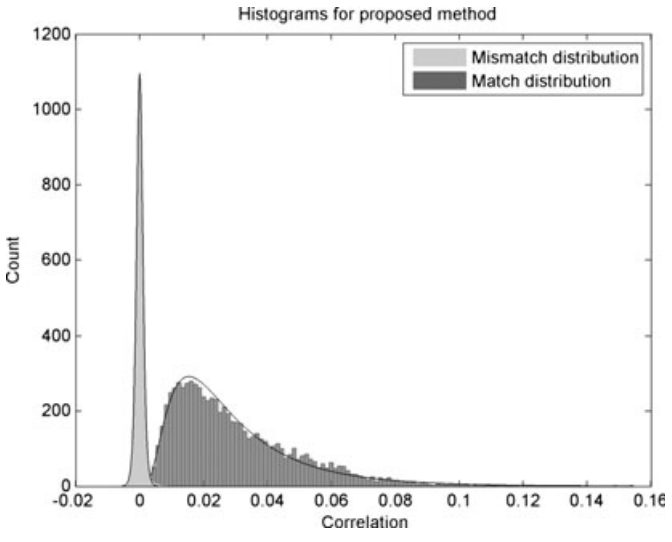

FIG. 2—*Probability density functions for matching (lognormal distribution) and mismatching (logistic distribution) data for the proposed method. There is a certain amount of overlap, but the separation is clear.*

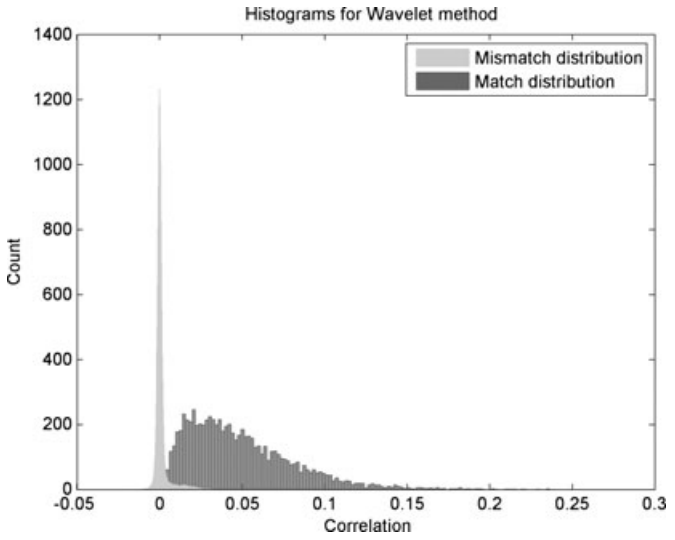

FIG. 3—*Histogram for the correlation values obtained from the wavelet method. We see the tails of the mismatching distribution is wider than in Fig. 2. This is mainly due to the large number of false positives from a single camera (Sony DSC-S930).*

function. These parameters can be estimated by nonlinear regression. The best fit, however, comes from a nonparametric model (e.g., kernel density estimation).

On the other hand, the correlation values for the matching pairs could be best described by a different type of distribution, namely by the lognormal distribution:

$$f(x|\mu,\sigma) = \frac{1}{x\sigma\sqrt{2\pi}}\exp\left(-\frac{(\ln x-\mu)^2}{2\sigma^2}\right) \qquad (14)$$

as the data were strongly skewed to the left. In this formula, μ and σ denote the mean and standard deviation of $\ln(x)$, respectively. In Fig. 2, the probability density function is presented for these suggested distributions. Finally, in Fig. 3, the histogram of correlation values is shown for the Wavelet method. It can be seen that the distribution is narrower for the proposed method, resulting in the better performance (lower EER). Upon inspection, we found that the wider tails in the wavelet distribution are almost exclusively owing to its bad performance on the Sony DSC-S930. Without this camera, the EER would be approximately 1.3%.

Conclusion

We have presented an alternative technique that can be used for the efficient extraction of PRNU patterns from images. The advantages are the simplicity of implementation, the reduced computation time (approximately 30% reduction), and the improved performance.

In the near future, this approach will be implemented in our open source program for camera comparison, NFI PRNUCompare (16).

Acknowledgments

We wish to thank the reviewers for their time, and our colleagues for their help making the photographs needed for this article.

References

1. Çeliktutan O, Sankur B, Avcibaş İ. Blind identification of source cell-phone model. IEEE Trans Inf Forensics Sec 2008;3:553–66.
2. Swaminathan A, Wu M, Liu KJR. Non-intrusive component forensics of visual sensors using output images. IEEE Trans Inf Forensics Sec 2007;1(2):91–106.
3. Gul G, Avcibas I. Source cell phone camera identification based on singular value decomposition. Proceedings of the first IEEE International Workshop on Information Forensics and Security, 2009; 171–5, DOI: 10.1109/WIFS.2009.5386459, http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=5377995 (accessed January 10, 2012).
4. Geradts Z, Bijhold J, Kieft M, Kurosawa K, Kuroki K, Saitoh N. Methods for identification of images acquired with digital cameras. Proceedings of SPIE, Enabling Technologies for Law Enforcement and Security (Progress in Biomedical Optics and Imaging); 2001 Feb; Boston, MA. Bellingham, WA: SPIE-International Society for Optical Engineering 2001;4232:505–12.
5. Lukáš J, Fridrich J, Goljan M. Digital camera identification from sensor pattern noise. IEEE Trans Inf Forensics Sec 2006;1:205–14.
6. van Houten W, Geradts Z. Source video camera identification for multiply compressed videos originating from YouTube. Digit Investig 2009;6:48–60.
7. Goljan M, Fridrich J. Camera identification from scaled and cropped images. Proceedings of SPIE, Electronic Imaging, Forensics, Security, Steganography, and Watermarking of Multimedia Contents X; 2008 Jan 26–31; San Jose, CA. SPIE-International Society for Optical Engineering 2008;6819:68190E-68190E-13.
8. Zhang C, Zhangh H. Digital camera identification based on curvelet transform. Proceedings of the 34th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2009); 2009 April 19–24; Taipei, Taiwan. Taipei, Taiwan: ICASSP, 2009;1389–92.
9. Perona P, Malik J. Scale-space and edge detection using anisotropic diffusion. Trans Pattern Anal Mach Intell 1990;12:629–39.
10. Goljan M, Fridrich J, Filler T. Large scale test of sensor fingerprint camera identification. In: Delp EJ, Dittmann J, Memon ND, Wong PW, editors. Proceedings of the SPIE; 2009 Jan 19; San Jose, CA. SPIE-International Society for Optical Engineering 2009;7254:725401–12.
11. Chen M, Fridrich J, Goljan M. Digital imaging sensor identification (further study). Proceedings of SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents X; 2007 Jan 29–Feb 1; San Jose, CA. SPIE-International Society for Optical Engineering 2007;6505:0P-0Q.
12. Gerig G, Kübler O, Kikinis R, Jolesz FA. Nonlinear anisotropic filtering of MRI data. IEEE Trans Med Imaging 1992;11:221–32.
13. Donoho DL, Maleki A, Shahram M. Wavelab 850, http://www-stat.stanford.edu/~wavelab/Wavelab_850/Contact.html (accessed October 26, 2010).
14. Donoho DL, Johnstone IM. Ideal spatial adaptation by wavelet shrinkage. Biometrika 1994;81(3):425–55.
15. Li CT. Source camera identification using enhanced sensor pattern noise. IEEE Trans Inf Forensics Sec 2010;5(2):280–7.
16. Van der Mark M, van Houten W, Geradts Z. NFI PRNUCompare, 2010, http://prnucompare.sourceforge.net (accessed October 26, 2010).

Additional information and reprint requests:
Wiger van Houten, M.Sc.
Northern Criminal Investigation Unit
Department of Digital Evidence
Schweitzerlaan 1
9728 NP Groningen
The Netherlands
E-mail: wawvanhouten@gmail.com

Appendix

Step 1 Set the maximum scale n (default: 3) and which diffusion function (Eq. [11]) should be used. Set $\lambda = 1/3$ and define the matrices g needed for convolution (eight in total):

$$\begin{pmatrix}0&1&0\\0&-1&0\\0&0&0\end{pmatrix},\begin{pmatrix}0&0&0\\0&-1&1\\0&0&0\end{pmatrix},\begin{pmatrix}0&0&0\\0&-1&0\\0&1&0\end{pmatrix},\begin{pmatrix}0&0&0\\1&-1&0\\0&0&0\end{pmatrix}$$

Read the input image I. Apply a median filter to this image.

Step 2 Set $I^t = I$, $I'^2 = \mathrm{I}$, $k = 1$

repeat n times:

Calculate the convolution of I^t with the first matrix g and store this in ΔI. Subsequently, calculate the value of K such that 90% of the intensity values in the gradient image ΔI occur below this value. Calculate c (Eq. [11]), depending on the diffusion function chosen. Set $I'^2 = I'^2 + \lambda c\ \Delta I$.

Clear variables $\forall I$ and c. Repeat for the other three matrices. Set $I^t = I'^2$ and $k = k + 1$

Step 3 Subtract the denoised image from the input image and zero-mean (9) the result M, as follows. First, subtract the column averages of M from M, giving M'. Then, subtract the row averages of M' from M'. This is the PRNU pattern.