Università degli Studi di Padova
Department of Information Engineering

Digital Forensics
Second Laboratory Report

Faccin Dario
ID Number: 1177736

Abstract

Digital visual media represent nowadays one of the principal means for communication. Lately, the reliability of digital visual information has been questioned, due to the ease in counterfeiting both its origin and content.

Digital image forensics is a research field which aims at validating the authenticity of images by recovering information about their history.

The main problem addressed in this report is the identification of the imaging device that captured the image.

Source identification

Image acquisition process

The light enters the imaging device through a system of optical lenses, which conveys it towards the imaging sensor. The imaging sensor is the heart of every digital camera, and it is composed of an array of photo detectors, each corresponding to a pixel of the final image, which transform the incoming light intensity into a proportional voltage.

Most cameras use CCD (Charged Coupled Device) sensors, but CMOS (Complementary Metal Oxide Semiconductor) imagers can also be found. To render color, before reaching the sensor the light is filtered by the Color Filter Array (CFA), a specific color mosaic that permits to each pixel to gather only one particular light wavelength (i.e. color).

The CFA pattern arrangement depends on the manufacturer, although Bayer's filter mosaic is often preferred. As a result, the sensor output is a mosaic of e.g. red, green and blue pixels arranged on a single layer.

To obtain the canonical 3-channels representation, the signal needs to be interpolated. Demosaicing algorithms are applied to this purpose; the missing pixel values in each layer are estimated based on the values of existing neighbors. Before the eventual storage, additional processing is performed, such as white balance, gamma correction, and image enhancement.

Finally, the image is recorded in the memory device. The following Figure 1 illustrates schematically the image acquisition process.

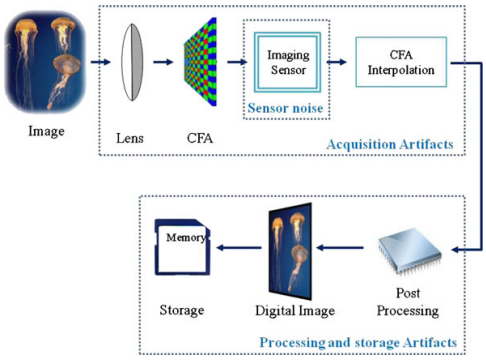


Figure 1. Image acquisition pipeline.

Camera identification

The described image acquisition pipeline is common for most of the commercially available devices. Each step is performed according to specific manufacturer choices and hence might depend on the camera brand and model.

This variation can be used to determine the type of camera from which a specific image was obtained. Indeed, each stage in the pipeline can introduce

imperfections in the final image or characteristic traits: lens distortion, chromatic aberration, pixel defects or CCD sensor imperfections, statistical dependencies related to proprietary CFA interpolation algorithms and other intrinsic image regularities.

These artifacts are statistically stable and can be considered as a signature of the camera type or even of the individual device.

Sensor imperfections

Imaging sensors have been shown to introduce various defects and to create noise in the pixel values. The sensor noise is the result of three main components, i.e. pixel defects, *Fixed Pattern Noise* (FPN), and *Photo Response Non Uniformity* (PRNU).

Pixel defects include point defects, hot point defects, dead pixels, pixel traps, and cluster defects, which reasonably vary across different sensors, independent on the specific camera model.

FPN and PNRU are the two components of the so-called pattern noise, and depend on dark currents in the sensor and pixel non-uniformities, respectively. Hence, they are independent on the image content but closely related to the physical characteristics of each single sensor.

The pattern noise extracted from images taken by the same camera are more correlated than those extracted from different cameras.

Photo Response Non Uniformity

Photo Response Non Uniformity (PRNU) is caused by the different sensitivity of the sensors to the light. This behavior is due to the manufacturing process and does not depend on the external temperature or acquisition time.

The resulting image can be described as following:

$$\mathbf{I} = \mathbf{I}^{(0)} + \mathbf{I}^{(0)}\mathbf{K} + \mathbf{\Theta} \tag{1}$$

where $\mathbf{I}^{(0)}$ is the ideal sensor output (without noise), $\mathbf{K}$ is the PRNU fingerprint of the camera and $\mathbf{\Theta}$ accounts for all the other types of noise.

Pattern noise can be estimated by taking the difference between an image $\mathbf{I}$ and its denoised version:

$$\mathbf{W_I} = \mathbf{I} - F\left(\mathbf{I}^{(0)}\right) \tag{2}$$

where $\mathbf{W_I}$ is called *residual noise* and F is a *denoising filter*.

PRNU fingerprint estimation

Let $\hat{\mathbf{I}}^{(0)}$ be the denoised version of $\mathbf{I}^{(0)}$.
We can rewrite Equation (2) as:

$$\begin{aligned}\mathbf{W_I} &= \mathbf{I} - \hat{\mathbf{I}}^{(0)} \\ &= \mathbf{I} - \hat{\mathbf{I}}^{(0)} + \mathbf{IK} - \mathbf{IK} \\ &= \mathbf{IK} + \mathbf{I}^{(0)} - \hat{\mathbf{I}}^{(0)} + \left(\mathbf{I}^{(0)} - \mathbf{I}\right)\mathbf{K} + \boldsymbol{\Theta}\end{aligned} \tag{3}$$

Now let $\boldsymbol{\Sigma} = \mathbf{I}^{(0)} - \hat{\mathbf{I}}^{(0)} + \left(\mathbf{I}^{(0)} - \mathbf{I}\right)\mathbf{K} + \boldsymbol{\Theta}$ be the noise independent from $\mathbf{IK}$. We can finally write:

$$\mathbf{W_I} = \mathbf{IK} + \boldsymbol{\Sigma} \tag{4}$$

Due to the random components related a specific image, the reference PNRU factor $\hat{\mathbf{K}}$ for a particular camera C is obtained can be estimated through Maximum Likelihood.

Given N images $\mathbf{I}_1, \ldots, \mathbf{I}_N$, we can reasonably assume that $\boldsymbol{\Sigma}[i]_1, \ldots, \boldsymbol{\Sigma}[i]_N$ for each pixel i are white Gaussian noise with variance σ^2. The energy of the PRNU $\mathbf{IK}$ is small compared to the noise term Θ, so we can also assume that $\boldsymbol{\Sigma}$ is independent of $\mathbf{IK}$.
For each $i = 1, \ldots, N$ we have

$$\frac{\mathbf{W}_i}{\mathbf{I}_i} = \mathbf{K} + \frac{\boldsymbol{\Sigma}_i}{\mathbf{I}_k}$$

The log-likelihood of observing $\dfrac{\mathbf{W}_i}{\mathbf{I}_i}$ given $\mathbf{K}$ is

$$L(\mathbf{K}) = -\frac{N}{2}\sum_{i=1}^{N}\log\left(\frac{2\pi\sigma^2}{(\mathbf{I}_i)^2}\right) - \sum_{i=1}^{N}\frac{\left(\dfrac{\mathbf{W}_i}{\mathbf{I}_i - \mathbf{K}}\right)^2}{\dfrac{2\sigma^2}{(\mathbf{I}_i)^2}} \tag{5}$$

The estimate is then obtained by computing the first order derivate of $L(\mathbf{K})$ with respect to $\mathbf{K}$ and solving for $\mathbf{K}$:

$$\frac{\partial L(\mathbf{K})}{\partial \mathbf{K}} = 0 \implies \hat{\mathbf{K}} = \frac{\sum\limits_{i=1}^{N}\mathbf{W}_i\mathbf{I}_i}{\sum\limits_{i=1}^{N}(\mathbf{I}_i)^2}$$

Computing the second order derivative is useful to obtain the Cramer-Rao lower bound and to infer what are the best images for the PRNU estimation.

$$\frac{\partial^2 L(\mathbf{K})}{\partial \mathbf{K}^2} = \frac{\sigma^2}{\sum\limits_{i=1}^{N}(\mathbf{I}_i)^2} \tag{6}$$

The luminance $\mathbf{I}_i$ should be as high as possible but not saturated, since saturated pixels carry no information on the PRNU factor.

Also $\text{var}(\hat{\mathbf{K}}) \sim \sigma^2$, therefore better estimates are obtained using smooth test images.

PRNU fingerprint detection

The estimated factor $\hat{\mathbf{K}}$ contains all components that are systematically present in every image. The most important are some weak artifacts of color interpolation, onsensor signal transfer and sensor design.

Such artifacts are not unique to the sensor and are shared among cameras of the same brand or cameras sharing the same imaging sensor design. The PRNU factors estimated from two different cameras may thus be slightly correlated, which would increase the false identification rate and decrease the reliability of camera identification.

To suppress these periodic traces in $\hat{\mathbf{K}}$, the mean of the rows and columns of the fingerprint is set to zero, and Wiener filtering is applied in the frequency domain after the maximum likelihood estimation.

These post operations increase the uniqueness of the fingerprint estimate among the same camera brand or model class.

This similarity of the estimated PRNU fingerprint can be measured by normalized cross correlation between $\mathbf{W_J}$ and $\hat{\mathbf{K}}\mathbf{J}$ as:

$$\rho = corr\left(\mathbf{W_J}, \hat{\mathbf{K}}\mathbf{J}\right) \tag{7}$$

If the image $\mathbf{I_J}$ is not taken by camera X, the maximum of the correlation ratio $\max\{\rho\}$ is expected to be close to zero. If the image $\mathbf{I_J}$ is taken by camera X, then the correlation should be significantly higher than zero.

However, it is not possible to set a reliable detection threshold for all camera devices because of the different resolutions and sensor types.

This issue has been solved using a *Peak-to-Correlation Energy* (PCE) ratio:

$$\text{PCE} = \frac{\rho_{\text{peak}}^2}{\frac{1}{|s|-|\epsilon|}\sum_{s \notin \epsilon} \rho_s^2} \tag{8}$$

where ρ_{peak} is the supremum of the normalized cross correlation between $\mathbf{W_J}$ and $\hat{\mathbf{K}}\mathbf{J}$ and s is the map to all entries of ρ. ϵ represents a small, centered region around ρ_{peak}, whereas $|s| - |\epsilon|$ is the total number of entries outside ϵ.

Results

The MATLAB code provided for this assignment performs the PRNU estimation and detection.

Among the returned values, the most important are the *Peak-to-Correlation Energy* (PCE) and the *Probability of False Alarm* (P_FA).

One tunable parameter is `windowsize`: this parameter allows the user to choose the span, in pixels, to consider for both the PRNU estimation and then detection.

For the data set provided with the code we have 6 flat-field images on which we estimate the PRNU fingerprint and 11 images whose origin is to predict.

Before running the script, we analyzed the EXIF information stored in both the flat-field and natural images. The flat-field images are taken by a *LG V30 (H930)*, while the natural images are taken by five different devices: *OnePlus 6T (A6013)*, *OnePlus 5 (A5000)*, *LG V30 (H930)*, *OnePlus 6 (A6003)*, *Xiaomi Mi A2 Lite*.

The following Table 1 shows the results of the code running on the provided data set:

Image	Acquisition Device	PCE (512)	PCE (2048)	P_{FA} (512)	P_{FA} (2048)
`nat_01`	OnePlus 6T (A6013)	6.0638	-1.0006	0.0069	0.8414
`nat_02`	OnePlus 5 (A5000)	0.0892	0.1314	0.3826	0.8535
`nat_03`	LG V30 (H930)	37.2846	681.8796	5.1044e-10	1.3036e-150
`nat_04`	OnePlus 6 (A6003)	-0.1512	3.0514	0.6513	0.0403
`nat_05`	LG V30 (H930)	46.7558	993.4452	4.0203e-12	2.3880e-218
`nat_06`	LG V30 (H930)	124.5864	1.3225e+03	3.1345e-29	7.2932e-290
`nat_07`	Xiaomi Mi A2 Lite	-4.8861	-0.5088	0.9865	0.7622
`nat_08`	Xiaomi Mi A2 Lite	0.0036	2.2319	0.4760	0.0676
`nat_09`	OnePlus 6T (A6013)	-0.2413	-0.0808	0.6884	0.6119
`nat_10`	OnePlus 6 (A6003)	0.0132	0.0023	0.4542	0.4808
`nat_11`	LG V30 (H930)	26.7829	156.8201	1.1382e-07	2.8018e-36

Table 1. Results obtained for the default data set, for two parameters `windowsize`.

From Table 1 we see that the images associated with an "high" PCE value and "low" probability of false alarm are the ones which are taken by the same device used for the PRNU fingerprint estimation.

Increasing the tunable parameter `windowsize` did improve the accuracy in the detection: while in the first case (512 pixels) one image (`nat_01`) has non-negligible PCE (6.0638), in the second case (2048 pixels) it is instead negligible (-1.0006). On the other hand the main drawback is the increased complexity and therefore processing processing time increases.

These results are more clear when looking at Figure 2 and Figure 3, which present the pair (PCE, P_{FA}) for each image, in both linear and logarithmic scale.

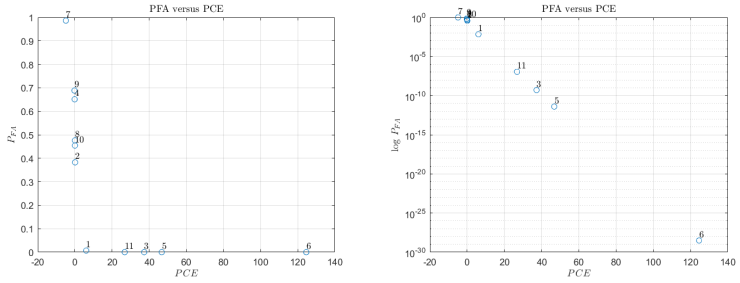


Figure 2. Results obtained for the default data set, for `windowsize=512`.

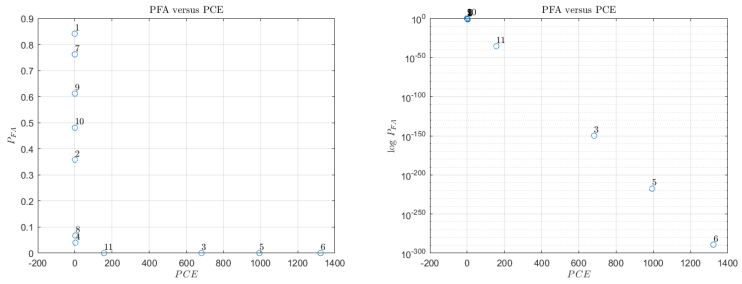


Figure 3. Results obtained for the default data set, for `windowsize=2048`.

The previous analysis has then been repeated using flat-field images taken by a *Canon EOS M50* and test images as before, plus two images from the incriminated camera.

Image	Acquisition Device	PCE (512)	P_{FA} (512)
`nat_01`	OnePlus 6T (A6013)	-0.1751	0.6622
`nat_02`	OnePlus 5 (A5000)	0.0236	0.4389
`nat_03`	LG V30 (H930)	-3.0625	0.9599
`nat_04`	OnePlus 6 (A6003)	-0.8005	0.8145
`nat_05`	LG V30 (H930)	1.8642	0.0861
`nat_06`	LG V30 (H930)	-0.0491	0.5876
`nat_07`	Xiaomi Mi A2 Lite	-0.0687	0.6033
`nat_08`	Xiaomi Mi A2 Lite	4.6016	0.0160
`nat_09`	OnePlus 6T (A6013)	2.0962	0.0738
`nat_10`	OnePlus 6 (A6003)	1.5389	0.1074
`nat_11`	LG V30 (H930)	0.1089	0.3707
`nat_13`	Canon EOS M50	400.2607	2.4163e-89
`nat_13`	Canon EOS M50	828.0930	2.1049e-182

Table 2. Results obtained for the modified data set, for `windowsize=512`.

From Figure 4 it is possible to see graphically how better the algorithm

performs in this situation, despite the low value for `windowsize` (512 pixels).

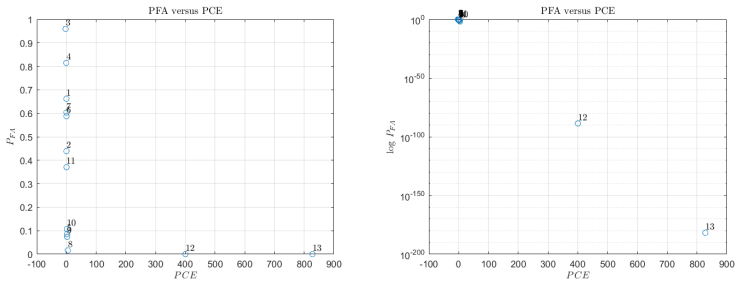


Figure 4. Results obtained for the modified data set, for `windowsize=1024`.

The following Figure 5 shows how the parameter `windowsize` affects the computational time of the PRNU estimation and detection:

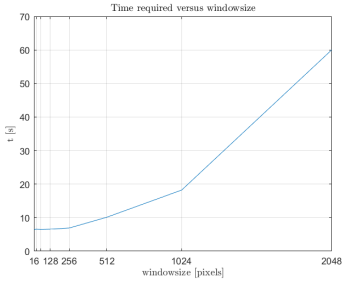
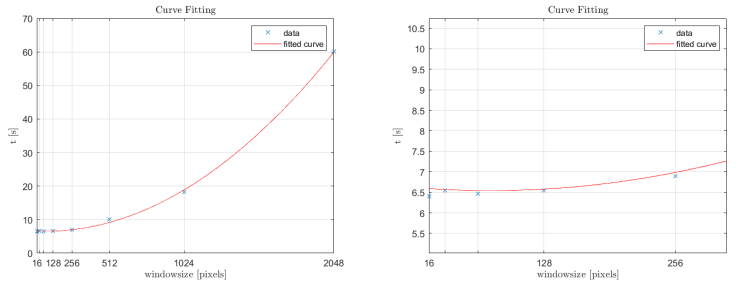


Figure 5. Time required for different `windowsize`.

Regarding the algorithm complexity, the *Curve Fitting Toolbox* provided by MATLAB has been used to roughly estimate whether the behavior is exponential or no.

The following Figure 6 shows how the software interpolated the data we have:



(a) Full fitting. **(b)** Zoom.

Figure 6. Results of the fitting.

The algorithm complexity can be modeled both as quadratic or exponential function:

$$C_{PRNU}(x) = \left(1.373 \cdot 10^{-5}\right) x^2 + 6.622 \tag{9}$$

$$C_{PRNU}(x) = 5.701 e^{0.00115x} \tag{10}$$

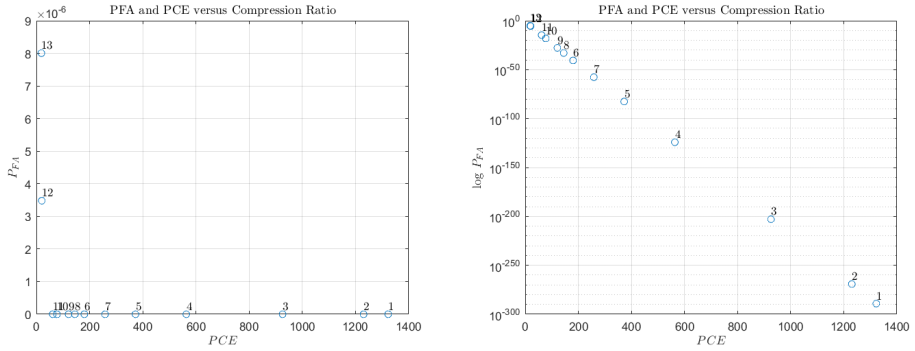
where x is the `windowsize`.

Now we evaluate the detection performance using the one single image but with different compression ratios.

Despite the original image being already compressed in a lossy form (JPEG standard), for the sake of the analysis we consider it as uncompressed. Therefore we would expect better performance if the original image is in an uncompressed format (RAW).

What we expect is that the PRNU detection works linearly with the quality of the image, meaning that as the image quality decreases the detection would degrade.

In Figure 7 the test image (`nat_05`) versions are numbered in increasing order of compression ratio (1: original "uncompressed" image, 13: most compressed).



(a) Results in linear scale. **(b)** Results in logarithmic scale.

Figure 7. Results obtained for image `nat_05`.

References

[1] Chen M., Fridrich J., Goljan M., Lukáš J., *Determining Image Origin and Integrity Using Sensor Noise*. 2008.

[2] Dirik A. E., Karaküçük A., *Forensic use of photo response non-uniformity of imaging sensors and a counter method*. 2014.

[3] Dugelay J., Redi J., Taktak W., *Digital image forensics*. 2010.

[4] Milani S., *Digital Forensics: course lectures*. 2018.