




Digital camera identification based on analysis of optical defects

Jarosław Bernacki¹ 

Received: 12 December 2018 / Revised: 9 June 2019 / Accepted: 2 September 2019 /
Published online: 4 December 2019
© The Author(s) 2019

Abstract

In this paper we deal with the problem of digital camera identification by photographs. Identifying camera is possible by analyzing camera's sensor artifacts that occur during the process of photo processing. The problem of digital camera identification has been popular for a long time. Recently many effective and robust algorithms for solving this problem have been proposed. However, almost all solutions are based on state-of-the-art algorithm, proposed by Lukás et al. in 2006. Core of this algorithm is to calculate the so-called sensor pattern noise based on denoising images with wavelet-based denoising filter. Such technique is very efficient, but very time consuming. In this paper we consider tracing cameras by analyzing defects of their optical systems, like vignetting and lens distortion. We show that analysis of vignetting defect allows for recognizing brand of the camera. Lens distortion can be used to distinguish images from different cameras. Experimental evaluation was carried out on 60 devices (compact cameras and smartphones) for a total number of 12 051 images, with support of the Dresden Image Database. Proposed methods do not require denoising images with wavelet-based denoising filter what has a significant influence for speed of image processing, compared with state-of-the-art algorithm.

Keywords Hardwaremetry · Camera recognition · Sensor identification · Photo response non uniformity · Distortion · Vignetting · Privacy · Digital forensics

1 Introduction

The popularity of photography is obvious. Lots of people take photos with their smartphones or cameras and immediately share them in social media. It is clear that tracing cameras may expose users' privacy to a serious threat. Problem of tracing digital cameras has been

✉ Jarosław Bernacki
jaroslaw.bernacki@outlook.com

¹ Częstochowa University of Technology, al. Armii Krajowej 36, 42-200, Częstochowa, Poland

popular for a long time. Tracing a camera is understood as a recognition of camera's sensor. In [11] process of searching for characteristic features for identifying the camera is called *hardwaremetry*. According to [13, 31] each camera leaves some specific and unique traits in the images that makes it possible to trace the camera and could serve as a "camera fingerprint". Such fingerprint is most often understood as pixel artifacts resulting of sensor imperfections or defects of optics. One of the classic and state-of-the-art algorithms for camera recognition was presented by Lukás et al. in [31]. For each camera it is determined the so-called sensor pattern noise that acts as an unique camera fingerprint. This sensor pattern noise is calculated by averaging noise obtained from multiple camera's images using a denoise filter. According to the authors, efficiency of the algorithm depends on denoising filter used for processing the images and experimental evaluation showed that wavelet-based denoising filter achieves best results. Such approach is very effective and the recognition of the camera is at very high level. However, using the wavelet-based denoising filter is very time consuming [30]. Typical time for denoising 12 megapixel photo takes about two minutes; time for denoising 24 megapixel photo takes three to even four minutes. It is clear that this issue makes usage of this approach for large sets of images impractical. Therefore, there is a motivation to consider methods that will be even less accurate, but much faster.

In this paper we discuss quite different approach in order to recognize the camera. We are not considering afore mentioned noise that can be extracted from images but typical defects of optical system, like lens distortion and vignetting.

Vignetting is a kind of fault that occurs due to optical defects or sensor imperfections [10, 34]. It reveals as a reduction of image brightness at the edges of the image. Therefore, vignetted images have visually darker corners than rest of an image. Such defect is popular in lots of digital cameras (especially compacts and digital single lens reflectives). Types of vignetting are precisely described in [10]. This topic has attracted many researchers, therefore a lot of algorithms [6, 22–24] and patents [27] for vignetting correction are developed. Lens distortion is a deviation from rectilinear projection [5, 32]. This phenomenon causes straight lines in the picture to become curved. It is seen in the images as differences in magnification of the image depending on its distance from the optical axis [14].

We analyze these defects in order to recognize camera's brand. The goal is to reduce the time for processing the image in comparison to Lukás et al.'s method. To the best of our knowledge, nobody before has tried to use vignetting or lens distortion for tracing the camera.

1.1 Contribution

Contribution of this paper is twofold. First, we analyze the vignetting defect in order to identify brand of the camera. We examine reduction of photo brightness at the edges of a set of images with the same frame perspective. We experimentally show that there are some tendencies of underexposed areas in the image edges that might be helpful to recognize camera's brand or even camera's model. Secondly, we show that analyzing image distortion can also be used for distinguish two cameras. Proposed observations are less efficient in camera's brand recognition compared with Lukás et al.'s algorithm [31] but significantly faster, what makes it practical to use them for processing large sets of images.

1.2 Organization of the paper

The paper is organized as follows. In Section 2 related and previous work is described. Section 3 describes the analysis of vignetting and lens distortion. In Section 4 experimental

evaluation is presented. Final section concludes this work and points some future research directions.

2 Related and previous work

The issue of tracing cameras is studied in various ways. One of the most popular and considered as state-of-the-art work on camera sensor recognition is Lukás et al.'s algorithm [31]. This algorithm is in detail described in Appendix. Authors proposed an algorithm for calculating camera's fingerprint which is based on a difference between image p and its denoised form $F(p)$.

In [2] we have proposed a fast method for camera tracing based on Peak signal-to-noise ratio. Method is very fast in comparison with [31], however classification efficiency is lower.

In [1] authors proposed using the k -means algorithm for managing photo response non-uniformity (PRNU) patterns. Patterns are compared to each other using correlation and grouped by k -means algorithm. Therefore, similar patterns grouped over a cluster are considered as belonging to the same camera. Experiments were conducted for a database of 500 images. Images grouped within a cluster were with true positive rate (TPR) of 98% belonging to the particular camera. The idea of fingerprint clustering is also presented in [3].

In [20] analysis of JPEG compression is carried out. It is well-known that JPEG lossy compression generates a noise that impacts groups of pixels. However, authors show JPEG compression adds some specific artifacts to the final image and the exact implementation details may be used for identification.

In [17] an approach of counterfeiting characteristic features of the camera in order to produce an image “pretending” to be done by another camera is presented. The technique is described as photo-response nonuniformity fingerprint-copy attack. The goal is to obfuscate sensor pattern noise of a particular camera by “inserting” into it a sensor pattern noise of another camera. It is showed that this can be done by performing simple algebraic operations. Let us assume that \hat{K}_N is a camera fingerprint calculated of N images and J is a fingerprint of another camera whose we want to put into \hat{K}_N . Then we have $J' = J(1 + \alpha \hat{K}_N)$, where $\alpha > 0$ is a scalar defining fingerprint strength. Experimental results show that such “exchange” of camera fingerprints is very efficient, i.e. based on \hat{K}_N we can produce a counterfeit photo J' pretending that camera. Nearly the same proposition is described in [35, 41]. This technique has a serious disadvantage, because it may be impractical. It is required to have a representative image set of camera that we want to “exchange” a fingerprint with and affects the actual image (i.e., the stored information). Due to denoising images, this method is also very time consuming.

In [4, 26, 29] dead pixels, pixels traps, point/hot point defects and cluster defects were investigated in terms of camera recognition. Experimental results show that different cameras have a distinct pattern of defective pixels, hence in some cases, hot and dead pixels allow recognizing the sensor.

In [21] a similar method as in [31] is presented. A sensor fingerprint is considered as a white noise present on the images. Authors suggest using correlation to circular correlation standard as a test statistic, which may reduce false positive rate (FPR) of camera recognition. However in contrast to [31], authors examine proposed method on fragments of photos instead of “full” photos. The true positive rate of recognition was 95% (fragments of size 256x256px) and 99% (512x512px).

In [13] a technique based on cross-correlation analysis and peak-to-correlation-energy (PCE) ratio to identify the camera is proposed. Sensor pattern noise is calculated and the correlation detector with PCE ratio to measure the similarity between noise residuals is used. However, time performance is not examined.

In [18] a method for camera identification using correlation is presented. Authors consider existing database with different cameras' fingerprints and calculate correlation coefficient of a fingerprint of a new camera for comparison. This approach is clearly based on Lukás et al.'s algorithm, moreover authors do not describe, how database of fingerprints is gained.

In [39, 40] a gradient technique for vignetting correction is described. It is also pointed that vignetting can be described by natural image statistics. In [23–25] polynomial models for vignetting correction are proposed.

In [32] two methods for calculating lens distortion in order to camera calibration are presented. First method is based on look-up-tables (LUT) of focal length and lens distortion. Second method uses relationships between some feature points found in image. Calculation of lens distortion is done by algebraic operations.

Work [28] presents a method for camera calibration and radial distortion correction. Radial distortion can be calculated by two distorted images. Advantage of this method is that no knowledge about camera's intrinsic parameters nor scene structure is required.

In [42] an innovative technique using Game Theory approach for identifying digital camera is discussed. The aim is to detect fingerprint-copy attack, when an adversary uses copy of original camera's fingerprint. Therefore, this problem is represented as an interplay between sensor-based camera identification and the fingerprint-copy attack. A Bayesian game is used for analyzing differences between original camera's fingerprint and the fingerprint-copy. The Nash equilibrium is used to evaluate the efficacy of proposed method.

In [33] there is proposed an approach for identifying a camera with the use of enhanced Poissonian-Gaussian model. This model describes distribution of pixels in a RAW image. Cameras' fingerprints are represented as parameters of a statistical noise of considered model. Experiments are conducted with Dresden Image Database and also authors' own image set.

In [37] problem of image splicing is considered. It is assumed that image splicing can be detected by analyzing noise level in spliced parts. It turns that splicing parts have different level of noise, what causes noise inconsistencies between them. This in turn allow for detecting the activity of splicing. A noise level function (NLF) is used. Experimental evaluation confirms efficacy of the NLF estimation. Work [37] is an extension of [33], where inconsistencies of images spliced regions are examined. Due to limitations of standard solution for estimating noise variance of each region, authors propose to use scoring strategy. An image is divided into small patches and the noise variance is calculated by kurtosis concentration-based pixel-level noise estimation method. Then, a sample of the noise variance and an inhomogeneity score of each region is fitted by a linear function. Experimental results confirmed efficacy of proposed method.

In [38] the problem of managing a large database of camera fingerprints is considered. Cameras' fingerprints are represented as matrices of the noise whose resolution is equal to camera's produced image size. A brute-force searching a specified fingerprint in a large database of N fingerprints takes $O(nN)$, where n is the number of pixels in each fingerprint. Therefore, the goal is to reduce this time. In considered paper a fast search algorithm is proposed. Algorithm extracts a digest of the query fingerprint of the 10,000 fingerprint values and approximately matches their positions with positions of pixels in the digests of

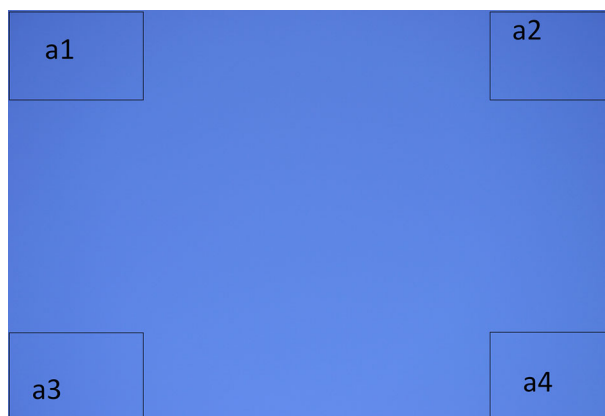


Fig. 1 Example of image of a plain surface. The pixel intensity in the middle of the image is 94; average pixel intensities in corners a_1 , a_2 , a_3 , a_4 are respectively: 74, 72, 84, 85

all database fingerprints. In the worst case, the complexity of this algorithm still could be the same as the database size, but in practice it is much faster. Experiments showed that for two-megapixel fingerprints searching takes 0.2 of second. However, this approach has some serious limitations. Nowadays, cameras' image sensors are much bigger than two-megapixels therefore search time will again increase. Secondly, this approach deals with searching the existing set of cameras' fingerprints, Therefore it is required to have such fingerprint set. This is still not practical due to big sizes of recent cameras sensors, where for example for 24-megapixel sensor, the fingerprint still must be a matrix with a corresponding size.

In [16] a method for camera identification is discussed. Calculating camera's fingerprint is in the similar spirit as in [31]. Evaluation is made using the Peak to Correlation Energy ratio (PCE). Experiments are very representative, images are utilized from the popular on-line image sharing site Flickr. Tests included more than million images of 6896 cameras covering 150 models. In [15] the same problem is solved by using support vector machines (SVM) with decision fusion techniques.

3 Analysis of vignetting and lens distortion

In this section we consider artifacts of vignetting and lens distortion in order to recognize camera's brand. In both cases an image is represented according to the \mathcal{RGB} model as $M \times N$ by 3 data array that defines red, green and blue color components for each individual pixel.

3.1 Vignetting-CT algorithm

As mentioned in the Introduction, vignetting is a defect depending on reduction of brightness at the image frame, usually in its corners. The easiest way to observe the presence of vignetting is a photo of plain surface (Fig. 1).

We propose a procedure called Vignetting-Camera Tracing (Vignetting-CT) for calculating the differences of pixel intensities in image corners (Algorithm 1).

Algorithm 1 Vignetting-CT.**Input:** Image G of size $M \times N$ **Output:** \hat{d} (mean value of pixel intensities in image corners)

```

1:  $c := G(\mathcal{R})$  //take one color channel of input image  $G$ 
2:  $f := \text{median\_filter}(c)$ 
3:  $S := |c - f|$ 

4:  $m := \lfloor \delta \cdot M \rfloor$ 
5:  $n := \lfloor \delta \cdot N \rfloor$ 

6:  $\hat{s}_{a_1} := \frac{\sum_{i=1}^m \sum_{j=1}^n s_{ij}}{mn}$  //  $s_{ij}$  are values of  $S$ 
7:  $\hat{s}_{a_2} := \frac{\sum_{i=1}^m \sum_{j=N-(n-1)}^N s_{ij}}{mn}$ 
8:  $\hat{s}_{a_3} := \frac{\sum_{i=M-(m-1)}^M \sum_{j=1}^n s_{ij}}{mn}$ 
9:  $\hat{s}_{a_4} := \frac{\sum_{i=M-(m-1)}^M \sum_{j=N-(n-1)}^N s_{ij}}{mn}$ 

10:  $\hat{d} := \frac{1}{4} \sum_{k=1}^4 \hat{s}_{a_k}$ 

11: return  $\hat{d}$ 

```

Vignetting-CT algorithm is very simple and comes to divide an image into four “small” parts at its corners and to calculate mean values of pixel intensities expressed in \mathcal{RGB} notation. This algorithm can be performed for any color channel, but we propose to process red (\mathcal{R}) color channel (line 1). Next step is median filtering of this channel (line 2) and calculating the *residuum* S defined as an absolute difference of the color channel and its median-filtered form (line 3). The δ parameter defines size of image parts that are to be analyzed (lines 4, 5). We propose to use $\delta = 0.05$. Then, mean values of pixel intensities in image corners are calculated (lines 6–9). Finally, we compute mean value of pixel intensities \hat{d} in corners of residuum S (line 10). We propose the \hat{d} value to use as a fingerprint for recognizing camera’s brand. Sample division of image frame is presented in Fig. 1.

Worth noticing that operating on S reveals the vignetting even, if image is not blank. We have inspected in detail pixel intensities of residuum S and it turns that in most images average pixel intensity of whole S is brighter than for considered corners. Sample image is presented in Fig. 2. Sample values for some photos from Nikon D70s (1) are presented in Table 1.

We propose to use median filter instead of wavelet-based denoising filter that is mainly used for denoising images [18, 21, 31]. Median filtering is noticeably faster than wavelet-based denoising. Vignetting-CT algorithm is computationally effective, its complexity is $O(mn)$, where m and n define the size of image parts for calculating pixel intensities. Due to small values of m and n , calculations are carried out very quickly.

3.2 Lens distortion

Distortion is a kind of defect of changing geometry of the image transmitted by lens to camera’s photosensitive sensor, especially closer to the corners and edges of the image frame. Essential for calculating distortion is the use of *calibration grid* which consists of crossing vertical and horizontal lines [5]. Such grid is used to compare it with lines in the image.

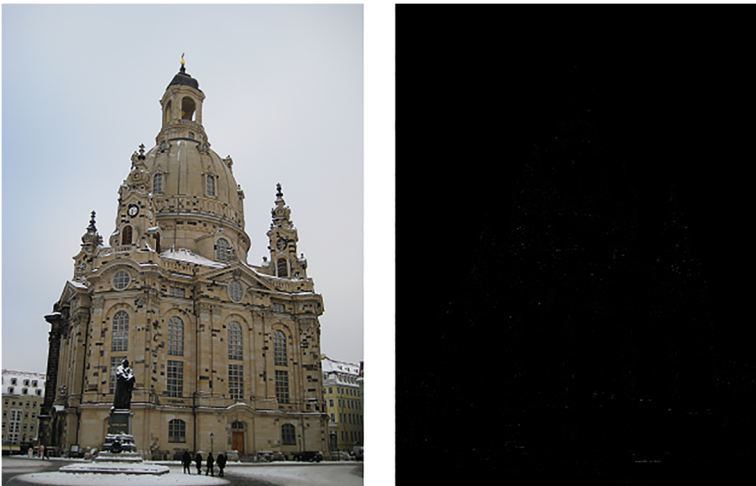


Fig. 2 Sample image and its residuum S calculated as absolute difference of an image and its median-filtered version. Average value of pixel intensities in S is 0.6, intensities in corners in S : $\hat{s}_{a_1} = 0.36$, $\hat{s}_{a_2} = 0.49$, $\hat{s}_{a_3} = 0.34$, $\hat{s}_{a_4} = 0.54$

If the lines in the image cover with the grid, there is no distortion in the photo. Otherwise, there is a *barrel* or *pincushion* distortion. Barrel distortion appears when image magnification decreases with distance from the optical axis. It often appears in pictures taken with wide-angle lenses. Image elements look as they were bent outside the image. Similarly, in pincushion distortion image magnification increases with distance to the optical axis. An image seems to be squeezed inside the center of the frame. A sample photo with visible barrel distortion is presented in Fig. 3.

Table 1 Sample vignetting values for Nikon D70s (1) in residuum S

Image	Average pixel intensities in S	\hat{s}_{a_1}	\hat{s}_{a_2}	\hat{s}_{a_3}	\hat{s}_{a_4}
1.jpg	0.59	0.34	0.32	0.40	0.41
2.jpg	0.69	0.44	0.43	0.46	0.45
3.jpg	0.31	0.22	0.18	0.24	0.15
4.jpg	0.28	0.23	0.24	0.24	0.26
5.jpg	0.45	0.25	0.29	0.41	0.28
6.jpg	0.45	0.15	0.09	0.27	0.40
7.jpg	0.34	0.19	0.19	0.25	0.15
8.jpg	0.47	0.22	0.19	0.23	0.28
9.jpg	0.24	0.23	0.23	0.22	0.22
10.jpg	0.31	0.20	0.26	0.21	0.19
11.jpg	0.40	0.30	0.31	0.33	0.35
12.jpg	0.36	0.27	0.26	0.27	0.26
13.jpg	0.65	0.46	0.38	0.58	0.55
14.jpg	0.54	0.38	0.37	0.39	0.41

The closer to 0, the darker area in the image

Fig. 3 Image with distortion and the grid of horizontal and vertical lines. Red lines denote distorted points



In literature there are many models that describe lens distortion. Most common are polynomial models [7, 14]. We propose a model defined as (1). This model is a simplified version of Brown's model [7].

$$p_u = p_d(1 + kr^2) \quad (1)$$

where:

- $p_u(x_u, y_u)$ – undistorted image point;
- $p_d(x_d, y_d)$ – distorted image point;
- k – distortion parameter;
- $r = \sqrt{(x_d - x_u)^2 + (y_d - y_u)^2}$.

We propose to calculate the distortion parameter k for a set of images of different cameras and check if there are some tendencies that might be helpful to be used in order to recognize a specific camera. The p_u and p_d can be determined by using software (for example Hugin Photo Stitcher [19]) or manually. Knowing distorted and undistorted points, this procedure leads for solving (1), where k is unknown. Of course, for a set of distorted and corresponding undistorted points there will be the same number of k values that might be different. In such case we propose to average of all k values.

Let us consider a simple toy example in which we show the reasoning of calculating the value of distortion.

Example 1 (Toy example) Suppose that we have coordinates of the following set of distorted p_d and corresponding undistorted p_u points:

$$\begin{aligned} p_{d1} &= (100, 80), p_{u1} = (110, 90) \\ p_{d2} &= (210, 100), p_{u2} = (200, 110) \\ p_{d3} &= (215, 177), p_{u3} = (217, 184) \end{aligned}$$

We calculate r_1 , r_2 and r_3 .

$$r_1 = \sqrt{(x_{d1} - x_{u1})^2 + (y_{d1} - y_{u1})^2} = \sqrt{(100 - 110)^2 + (80 - 90)^2} = 14.14$$

Similarly, $r_2 = 14.14$ and $r_3 = 7.28$

Inserting into (1) and calculating the k values for p_{d1} and p_{u1} :

$$100 = 110(1 + k \cdot 14.14)$$

$$k_{1(1)} = -0.09$$

$$80 = 90(1 + k \cdot 14.14)$$

$$k_{1(2)} = 0.07$$

Similarly, for next pair of points p_{d2} , p_{u2} and p_{d3} , p_{u3}

$$k_2 = -0.003 \text{ and } 0.007; k_3 = 0.001 \text{ and } -0.005$$

Finally, we calculate the mean value of all k values:

$$k = \frac{-0.09 + 0.07 + (-0.003) + 0.007 + 0.001 + (-0.005)}{6} = -0.003$$

Thus, the $k = -0.003$ is the searched parameter.

Above procedure should be performed for all images separately. We propose to consider the k parameter as unique that might be used to distinguish cameras. Due to simplicity, proposed procedure is fast and can be easily implemented.

4 Experimental verification

In this section we compare results of recognizing cameras' brands by analyzing \hat{d} value of Vignetting-CT and k distortion parameter with Lukás et al.'s algorithm [31] in terms of efficiency of classification and time performance. Details of Lukás et al.'s algorithm are recalled in the [Appendix](#). For evaluation of this algorithm, we use original authors' MATLAB implementation [30]. Both Vignetting-CT algorithm and script for calculating distortion parameter k are implemented in MATLAB.

We use the *accuracy* (ACC), defined in the standard way as an evaluation statistic:

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}},$$

where TP/TN denotes “true positive/true negative”; FP/FN stands for “false positive/false negative”. TP denotes number of cases correctly classified to a specific class; TN are instances that are correctly rejected. FP denotes cases incorrectly classified to the specific class; FN are cases incorrectly rejected.

4.1 Devices

Experiments are conducted on two datasets. First dataset contains images from popular smartphones (further called “smartphones dataset”). We have used 264 JPEG images from 12 smartphones. Used smartphones include: Apple iPhone 6, Asus ZenFone 2, HTC One M9, Huawei P8, LG G3, LG G4, Lumia 1020, Lumia 1520, Samsung Galaxy Note 4, Samsung Galaxy S6, Sony Xperia Z3 and Sony Xperia Z3+. All devices contain CMOS sensors. Second dataset include images from the Dresden Image Database [9]. This database consists of tens of thousands images made by different cameras and is often used for research [8, 12]. We have used 11787 JPEG images of 48 cameras. Used cameras include: Agfa DC 733s, Agfa DC 830i, Agfa Sensor 505, Agfa Sensor 530s, Canon Ixus 55, Canon Ixus 70 (3 devices), Casio EX Z150 (5 devices), Kodak M1063 (5 devices), Nikon CoolPix S710 (5 devices), Nikon D70 (2 devices), Nikon D70s (2 devices), Nikon D200 (2 devices), Olympus

Table 2 Confusion matrix of model recognition, Vignetting-CT, smartphones dataset, ACC = 0.62

		A1	A2	H1	H2	L1	L2	L3	L4	S1	S2	S3	S4
Apple iPhone 6	A1	69.2	*	*	*	7.7	*	*	*	*	15.4	7.7	*
Asus ZenFone 2	A2	*	75.0	*	*	25.0	*	*	*	*	*	*	*
HTC One M9	H1	*	*	75.0	*	*	*	*	*	*	*	25.0	*
Huawei P8	H2	*	*	*	60.0	*	20.0	*	6.7	*	13.3	*	*
LG G3	L1	*	*	12.5	12.5	37.5	*	25.0	*	*	*	12.5	*
LG G4	L2	*	*	*	*	12.5	62.5	*	*	25.0	*	*	*
Lumia 1020	L3	*	12.5	*	*	*	*	50.0	25.0	*	12.5	*	*
Lumia 1520	L4	*	*	*	*	*	*	50.0	50.0	*	*	*	*
Samsung Note 4	S1	*	*	*	16.7	*	*	*	*	66.7	16.7	*	*
Samsung S6	S2	*	*	*	*	*	*	4.5	*	13.6	72.7	*	*
Sony Xperia Z3	S3	*	*	*	14.3	*	*	*	*	*	*	71.4	*
Sony Xperia Z3+	S4	*	*	*	11.1	*	11.1	*	*	*	*	22.2	55.6

The symbol * represents values smaller than 5%

1050SW (5 devices), Praktica DCZ5 (5 devices), Rollei RCP 7325XS (3 devices), Samsung L74 (3 devices) and Samsung NV15 (3 devices). In most cases, images were taken of the same image frames by different devices. All cameras in this dataset contain CCD sensors. In cases of both datasets all images are JPEG lossy compressed and come directly from cameras. We do not assume further processing of images, for example user's graphic processing.

4.2 Experiment I – brand identification by analyzing vignetting

We analyze influence of underexposed areas in the corners of images for recognizing the camera. Experiments are performed in the following way. The \hat{d} value is calculated for every image from each camera and mean value of \hat{d} of images from a specific camera is calculated. To classify a new image K , its \hat{d}_K is calculated. Obtained result is assigned to the closest mean \hat{d} from a specific camera and assumed as taken by this camera.

For smartphones dataset the \hat{d} value has been calculated for 22 images per device. Results of classification are presented in Tables 2, 3, 4, 5.

Table 3 Confusion matrix of brand recognition, Vignetting-CT, smartphones dataset, ACC = 0.72

		A1	A2	H1	H2	L1	L2	S1	S2
Apple	A1	69.2	*	*	*	7.7	*	15.4	7.7
Asus	A2	*	75.0	*	*	25.0	*	*	*
HTC	H1	*	*	75.0	*	*	*	*	25.0
Huawei	H2	*	*	*	60.0	20.0	6.7	13.3	*
LG	L1	*	*	6.3	6.3	56.3	12.5	12.5	6.3
Lumia	L2	*	10.0	*	*	*	80.0	10.0	*
Samsung	S1	*	*	*	*	*	*	85.7	*
Sony	S2	*	*	*	12.5	6.3	*	6.3	75.0

The symbol * represents values smaller than 5%

Table 4 Confusion matrix of model recognition, Lukás et al.'s algorithm, smartphones dataset, ACC = 0.84

		A1	A2	H1	H2	L1	L2	L3	L4	S1	S2	S3	S4
Apple iPhone 6	A1	81.0	*	*	*	*	*	*	*	*	*	*	*
Asus ZenFone 2	A2	*	90.0	*	*	*	*	*	*	*	*	*	*
HTC One M9	H1	*	*	85.0	*	*	*	*	6.0	*	*	*	*
Huawei P8	H2	*	*	*	87.0	*	*	*	*	*	*	*	*
LG G3	L1	*	*	*	*	88.0	*	*	*	*	*	*	*
LG G4	L2	*	*	*	*	*	90.0	*	*	*	*	*	*
Lumia 1020	L3	*	*	*	*	*	*	80.0	*	*	*	*	*
Lumia 1520	L4	*	8.0	*	*	7.0	*	*	77.0	*	*	7.0	*
Samsung Note 4	S1	*	*	*	*	*	*	*	*	82.0	*	*	*
Samsung S6	S2	*	*	*	6.0	*	*	*	*	*	85.0	*	7.0
Sony Xperia Z3	S3	*	*	*	*	*	*	*	*	*	*	85.0	*
Sony Xperia Z3+	S4	*	6.0	*	*	*	6.0	*	*	7.0	*	*	81.0

The symbol * represents values smaller than 5%

For Dresden Image Database the \hat{d} value has been calculated for at least 180 images per device. Results of brand classification are presented in Tables 6 and 7. For the clarity, we do not present results for model classification, due to the amount of 48 cameras.

Results presented in confusion matrices point that camera's brand classification is noticeably higher for Lukás et al.'s algorithm. In cases of two tested datasets the accuracy of brand classification is 84% for smartphones dataset and 83% for Dresden Image Database. Efficiency of Vignetting-CT algorithm is 72% for smartphones dataset and 52% for Dresden Image Database. Advantage of Lukás et al.'s algorithm is classification "stability", what is especially visible in confusion matrices of brand recognition.

It is worth saying that Lukás et al.'s algorithm achieves better performance in case of older devices with CCD sensors. In [2] it was shown that performance of this algorithm is lower for newer cameras with CMOS sensors. Nowadays, recent cameras have CMOS sensors instead of CCDs.

Table 5 Confusion matrix of brand recognition, Lukás et al.'s algorithm, smartphones dataset, ACC = 0.84

		A1	A2	H1	H2	L1	L2	S1	S2
Apple	A1	81.0	*	*	*	*	*	*	*
Asus	A2	*	90.0	*	*	*	*	*	*
HTC	H1	*	*	85.0	*	*	*	*	*
Huawei	H2	*	*	*	87.0	*	*	*	*
LG	L1	*	*	*	*	89.0	*	*	*
Lumia	L2	*	*	*	*	*	78.5	*	*
Samsung	S1	*	*	*	*	*	*	83.5	*
Sony	S2	*	*	*	*	*	*	*	83.0

The symbol * represents values smaller than 5%

Table 6 Confusion matrix of brand recognition, Vignetting-CT, Dresden Image Database, ACC = 0.52

		Ag	C1	C2	Ko	Ni	Ol	Pr	Ro	Sa
Agfa	Ag	70.0	*	*	*	5.0	*	*	*	25.0
Canon	C1	*	60.0	*	*	5.0	*	15.0	15.0	5.0
Casio	C1	4.0	4.0	48.0	*	*	*	*	24.0	20.0
Kodak	Ko	4.0	*	8.0	40.0	28.0	12.0	*	4.0	4.0
Nikon	Ni	3.6	3.6	1.8	3.6	52.7	*	*	1.8	32.7
Olympus	Ol	12.0	8.0	8.0	4.0	16.0	40.0	*	*	12.0
Praktica	Pr	8.0	*	12.0	8.0	4.0	*	28.0	*	40.0
Rollei	Ro	*	*	13.3	*	6.7	*	6.7	40.0	33.3
Samsung	Sa	*	*	*	*	6.7	*	*	*	93.3

The symbol * represents values smaller than 5%

4.3 Time performance

Data presented in Table 8 and Fig. 4 clearly indicate that Lukás et al.'s algorithm is defeated in terms of time of image processing. Vignetting-CT algorithm processes images in real time, while Lukás et al.'s takes at average about 90 seconds to process a single image. Of course, image processing time is dependent on the image resolution. Lower resolution images (e.g. 6 megapixels 3000x2000px) are processed with less than 1 minute, however, 24 megapixels images of resolution 6000x4000px are processed about 4 minutes. To sum up, the whole time for processing 12051 images took less than 40 minutes in case of Vignetting-CT algorithm and about 294 hours for Lukás et al.'s algorithm. Such poor time performance excludes usage of Lukás' algorithm for a mass scale. In [36] it was examined if the image processing time Lukás' algorithm could be decreased by processing small fragments of images. For this purpose 50×50 pixels fragments of photos were used, however the results of classification were not satisfactory.

The experiments were conducted on MSI GV62-7RD notebook with quad-core Intel Core i5-7300HQ processor with 24GB of RAM. It is worth mentioning that camera's fingerprint in Lukás et al.'s algorithm is stored as a matrix of size of camera's images. Authors'

Table 7 Confusion matrix of brand recognition, Lukás et al., Dresden Image Database, ACC = 0.83

		Ag	C1	C2	Ko	Ni	Ol	Pr	Ro	Sa
Agfa	Ag	90.0	*	*	*	*	*	*	6.0	*
Canon	C1	*	77.0	*	*	*	*	9.0	*	7.0
Casio	C1	6.0	*	77.0	*	*	13.0	*	*	*
Kodak	Ko	*	7.0	*	82.0	*	*	*	*	8.0
Nikon	Ni	*	*	*	*	81.0	*	*	*	*
Olympus	Ol	*	*	*	*	*	79.0	*	*	*
Praktica	Pr	*	7.0	*	*	*	*	87.0	*	*
Rollei	Ro	*	8.0	*	*	10.0	*	*	79.0	*
Samsung	Sa	*	*	*	*	6.0	*	*	*	92.0

The symbol * represents values smaller than 5%

Table 8 Processing time of smartphones dataset and Dresden Image Database (12051 photos in total)

Algorithm	Overall	Average (per one photo)
Vignetting-CT	39 minutes	0.2 seconds
Lukás	294 hours	90 seconds

implementation produces the fingerprint files as MATLAB *.mat files which are usually of weight at least 110 megabytes. It means that calculated fingerprints for two used datasets of over 12 thousand of images weigh about 1.2 terabyte.

4.4 Experiment II – comparison of lens distortion

We analyze lens distortion parameter k for images from different devices but for the same image frames. Script for calculating k parameter was written in MATLAB, but it can also be used the Hugin Photo Stitcher software [19] that measures the distortion. We have compared distortion results from MATLAB script with Hugin software and obtained results are the same. Analysis shows that despite photographing the same scene, distortion parameter appears to be different for various devices. Sample results are presented in Figs. 5 and 6. It is worth mentioning that different devices of the same camera model give different values of k parameter. Such example is presented in Fig. 5, where the same image frame of two different smartphones (Huawei P8 and Samsung S6) generates different distortion parameters. A similar situation is presented in Fig. 6, for two different devices of Nikon CoolPix S710 (Dresden Image Database).

Sample results of k distortion parameter are presented in Tables 9 and 10. Due to large number of devices and images, we present for the clarity only part of full results. It is visible that all photos taken by various devices have different distortion parameters. Therefore proposed approach gives information if a set of images was done by one or more cameras, however it can not be used to determine of what model or even brand was used. What is more, an advantage of proposed method is its speed, because distortion parameter k is

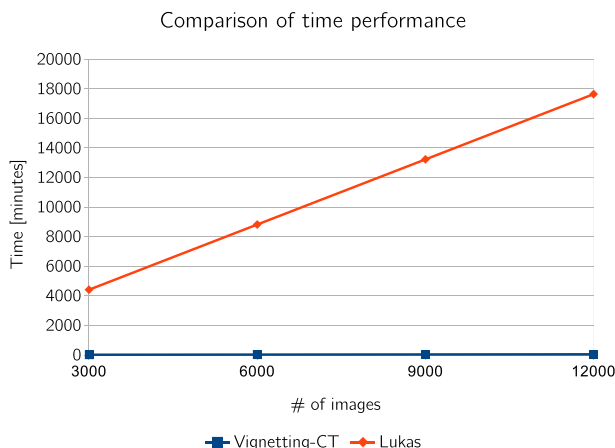
**Fig. 4** Comparison of time performance of Lukás et al.'s algorithm and Vignetting-CT for all images (12051)



Fig. 5 Lens distortion of images of the same frame, smartphones dataset. Image (1) (Huawei P8), distortion parameter $k = -0.6782$; image (2) (Samsung S6), distortion parameter $k = -0.2$

calculated in real time. Moreover, there are many applications that calculate and correct lens distortion in photos, thus there is no need to implement distortion algorithm manually. Of course proposed method can be useful in simple cases of comparing similar photos, however it may not be practical for sets of different images. One of the reason is that distortion parameter changes due to distance to the object, angle of view and focal length.

4.5 Summary

We have analyzed influence of vignetting and lens distortion defects to the problem of digital camera recognition. Experiments show that analysis of vignetting can be used for brand recognition. Compared to Lukás et al.'s algorithm, model recognition is lower, however Vignetting-CT algorithm beats Lukás et al.'s algorithm in terms of speed. Efficiency of brand recognition on smartphones dataset and Dresden Image Database is 72% and 52%, respectively, while Lukás et al.'s algorithm achieves 84% and 83%. However, an advantage of Vignetting-CT algorithm is processing images in real time, while Lukás et al. takes at average about 90 seconds per photo. Therefore, the Lukás et al.'s algorithm needed more



Fig. 6 Lens distortion of images of the same frame, Dresden Image Database. Image (1) (Nikon CoolPix S710 (device #2)), distortion parameter $k = -0.10323$; image (2) (Nikon CoolPix S710 (device #0)), distortion parameter $k = -0.01799$

Table 9 Sample values of k lens distortion parameter of the same image frame for different sample devices, smartphones dataset

Image	Apple iPhone 6	Lumia 1020	Samsung S6
1.jpg	0.266	−0.001	0.000
2.jpg	−0.027	0.077	−0.060
3.jpg	0.026	0.039	0.000
4.jpg	0.001	0.006	0.028
5.jpg	0.000	−0.056	−0.002
6.jpg	0.008	−0.013	0.032
7.jpg	0.000	0.001	0.006
8.jpg	−0.027	0.235	−0.042
9.jpg	−0.027	−0.030	−0.060
10.jpg	−0.001	−0.002	−0.017
11.jpg	−0.187	0.001	−0.060
12.jpg	−0.529	0.037	−0.012
13.jpg	−0.027	−0.082	−0.060
14.jpg	0.008	0.014	−0.060

than 290 hours to calculate cameras fingerprints, what excludes this algorithm of usage for a mass scale.

We have also experimentally shown that analysis of lens distortion can be useful to distinguish if a set of images of the same frame was taken by the same camera. Photos of the same frame but made with various cameras generate different distortion parameters. Of course, such approach may be not practical for camera recognition in case of images of different frames, however it can be useful for analysis of similar photos. The main disadvantage of distortion is its heterogeneity. Distortion changes due to distance to the object, focal length

Table 10 Sample values of k lens distortion parameter of the same image frame for different sample devices, Dresden Image Database

Image	Canon IXUS 70			Nikon D70	
	(0)	(1)	(2)	(0)	(1)
85.jpg	−0.002	−0.001	−0.012	−0.15	−0.23
86.jpg	−0.006	−0.034	−0.091	0.01	−0.07
87.jpg	−0.590	−0.119	0.001	−0.23	−0.361
88.jpg	0.17	0.42	0.207	−0.003	0.001
89.jpg	0.053	0.02	0.081	0.19	0.019
90.jpg	−0.05	−0.061	−0.018	−0.024	−0.039
91.jpg	0.634	0.52	0.784	0.499	0.506
92.jpg	0.501	0.222	0.36	0.91	0.68
93.jpg	−0.005	−0.04	−0.081	−0.2	−0.09
94.jpg	−0.62	−0.89	−0.7	−0.023	−0.601
95.jpg	−0.94	−0.88	−0.95	−0.55	−0.598
96.jpg	0.04	0.077	0.901	−0.004	−0.02
97.jpg	−0.001	0.004	−0.095	0.17	0.05
98.jpg	−0.062	−0.181	−0.56	−0.34	−0.21

Used three models of Canon IXUS 70 and two of Nikon D70

or angle of view, therefore there is probably no possibility to propose reasonable model that could be used for tracing cameras for a mass scale. Another limitation is that if there are no straight lines in the picture, distortion can not be determined.

5 Conclusion and future work

In this paper the problem of recognizing digital cameras by photographs was examined. Most popular solutions are based on denoising images with wavelet-based denoising filter and calculating so-called sensor pattern noise which averaged gives camera's fingerprint. We have proposed a novel approach for tracing digital cameras by analysis of vignetting and distortion defects. We have compared obtained results with state-of-the-art Lukás et al.'s algorithm. Experiments have shown that despite lower efficiency it is possible to recognize brand of the camera by analysis of vignetting defect. Our approach defeats the Lukás et al.'s algorithm in terms of image processing time. Proposed Vignetting-CT algorithm processes images in real time, while Lukás et al.'s algorithm needs at average 90 seconds to calculate the sensor pattern noise of one photograph. Moreover, Vignetting-CT algorithm does not require calculating cameras' fingerprints what is very time consuming. Analysis of distortion showed that images of different devices (also of the same model) generate different distortion parameters. Therefore, there is a possibility to distinguish if photos were taken by the same camera. This method is useful for a set of similar images and is very fast as well calculating distortion parameter is performed in real time.

Future works will concern further experiments with lens distortion analysis. It should be examined if it is possible to propose a model of distortion that could be used for reliable and more universal camera recognition. It would even be interesting to check if other optical defects like chromatic or spherical aberration would be useful to trace the camera's brand. Moreover, we are going to check efficiency of Vignetting-CT algorithm with classifiers based on Deep Learning or Convolutional Neural Networks approaches.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Appendix: Lukás et al.'s algorithm

Lukás et al. in [31] proposed an algorithm for identifying digital camera by produced images. This algorithm extracts a specific pattern called the photo-response nonuniformity noise (PRNU) which serves as a unique camera's identification fingerprint. Algorithm extracts noise from input image P by using a denoising filter F . Then, camera's fingerprint is calculated as $C = P - F(P)$.

Input: Image P in RGB of size $M \times N$;

Output: Matrix C of noise residual, size $M \times N$.

1. Calculate $K = \frac{P_R + P_G + P_B}{3}$;
2. Denoise all channels of input image P_R, P_G, P_B with filter F ;
3. Calculate mean $D = \frac{F(P_R) + F(P_G) + F(P_B)}{3}$;
4. Calculate matrix of noise residual: $C = K - D$.

where P_R , P_G and P_B are matrices of each component of RGB model of input P ; F is the denoising filter. For denoising, authors propose using the wavelet-based denoising filter.

Authors suggest to use at least 45 images from each camera to calculate the PRNUs for each image and then average obtained PRNUs, what as a result will give camera's fingerprint.

References

1. Baar T, van Houten W, Geradts Z (2012) Camera identification by grouping images from database, based on shared noise patterns. CoRR abs/1207.2641
2. Bernacki J, Klonowski M, Syga P (2017) Some Remarks about tracing digital cameras – faster method and usable countermeasure. In: Proceedings of the 14th International Joint Conference on e-Business and Telecommunications (ICETE 2017), pp 343–350
3. Bloy GJ (2008) Blind camera fingerprinting and image clustering. IEEE Trans Pattern Anal Mach Intell 3:30
4. Chapman GH, Thomas R, Thomas R, Koren Z, Koren I (2015) Enhanced correction methods for high density hot pixel defects in digital imagers
5. Claus D, Fitzgibbon AW (2005) A rational function lens distortion model for general cameras. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)
6. De Silva V, Chesnokov V, Larkin D (2016) A Novel Adaptive Shading Correction Algorithm for Camera Systems, Electronic Imaging, Digital Photography and Mobile Imaging XII, pp 1–5(5)
7. de Villiers JP, Leuschner FW, Geldenhuys R (2008) Centi-pixel accurate real-time inverse distortion correction, 2008 International Symposium on Optomechatronic Technologies. SPIE. <https://doi.org/10.1117/12.804771>
8. Deng Z, Gijsenij A, Zhang J (2011) Source camera identification using auto-white balance approximation. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp 57–64
9. (2008) Dresden Image Database. <http://forensics.inf.tu-dresden.de/ddimgdb/> Online; accessed 3 December 2018
10. Fuentes L et al (2015) Revisiting Image Vignetting Correction by Constrained Minimization of Log-Intensity Entropy Volume 9095 of the book series Lecture Notes in Computer Science (LNCS)
11. Galdi C, Nappi M, Dugelay J (2016) Multimodal authentication on smartphones: combining iris and sensor recognition for a double check of user identity. Pattern Recogn Lett 82:144–153
12. Gloe T (2012) Feature-based forensic camera model identification. LNCS Transactions on Data Hiding and Multimedia Security (DHMMS), to appear
13. Goljan M (2008) Digital camera identification from images - estimating false acceptance probability. In: Digital Watermarking, 7th International Workshop, IWDW 2008, pp 454–468
14. Goljan M, Fridrich J (2014) Estimation of lens distortion correction from single images. In: Proceedings volume 9028, media watermarking, Security, and Forensics 2014; 90280N
15. Goljan M, Fridrich J, Filler T (2009) Large scale test of sensor fingerprint camera identification, In: Proceedings of SPIE - The International Society for Optical Engineering, <https://doi.org/10.1117/12.805701>
16. Goljan M, Fridrich J, Filler T (2010) Managing a Large Database of Camera Fingerprints, In: Proceedings of SPIE - The International Society for Optical Engineering, <https://doi.org/10.1117/12.838378>
17. Goljan M, Fridrich JJ, Chen M (2011) Defending against fingerprint-copy attack in sensor-based camera identification. IEEE Trans Inf Forensics Secur 6(1):227–236
18. Hu Y, Li C-T, Lai Z, Zhang S Fast camera fingerprint search algorithm for source camera identification. In: Proceedings of the 5th International Symposium on Communications, Control and Signal Processing, ISCCSP 2012, Rome, Italy, 2–4 May 2012
19. Hugin Photo Stitcher <http://hugin.sourceforge.net/>, last access 4 December 2018
20. Julliard T, Nozick V, Talbot H (2016) Image Noise and Digital Image Forensics. Springer International Publishing, Cham, pp 3–17
21. Kang X, Li Y, Qu Z, Huang J (2012) Enhancing source camera identification performance with a camera reference phase sensor pattern noise. IEEE Trans Inf Forensics Secur 7(2):393–402
22. Khan MB, Nisar H, Ng CA, Lo PK (2016) A vignetting correction algorithm for Bright-Field microscopic images of activated sludge. In: 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)

23. Kordecki A, Bal A, Palus H (2016) Local polynomial model: a new approach to vignetting correction. In: Proc SPIE 10341 Ninth International Conference on Machine Vision
24. Kordecki A, Bal A, Palus H (2016) A study of vignetting correction methods in camera colorimetric calibration. In: Proc SPIE 10341 9th International Conference on Machine Vision
25. Kordecki A, Palus H, Bal A (2016) Practical vignetting correction method for digital camera with measurement of surface luminance distribution 10(8):1417–1424
26. Lanh TV, Chong K, Emmanuel S, Kankanhalli MS (2007) A survey on digital camera image forensic methods. In: Proceedings of the 2007 IEEE International Conference on Multimedia and Expo, ICME, 2007, pp 16–19
27. Lee SY, Cho HJ, Lee HJ (2016) Method for vignetting correction of image and apparatus therefor US 20160189353 A1
28. Li H, Hartley R (2005) A Non-iterative Method for Correcting Lens Distortion from Nine Point Correspondences, In: Proc OmniVision'05, ICCV-workshop
29. Li X, Shen H, Zhang L, Zhang H, Yuan Q (2014) Dead pixel completion of aqua MODIS band 6 using a robust m-estimator multiregression. IEEE Geosci Remote Sensing Lett 11(4):768–772
30. Lukás J, Fridrich JJ, Goljan M (2016) Matlab implementation
31. Lukás J, Fridrich JJ, Goljan M (2006) Digital camera identification from sensor pattern noise. IEEE Trans Inf Forensics Secur 1(2):205–214
32. Park S-W, Hong K-S (2001) Practical ways to calculate camera lens distortion for real-time camera calibration. Pattern Recogn 34:1199–1206
33. Qiao T, Retraint F (2018) Identifying individual camera device from RAW images, digital object identifier, vol 6, pp 78038–78054 IEEE Access, <https://doi.org/10.1109/ACCESS.2018.2884710>
34. Ray SF (2002) Applied photographic optics, 3rd ed. Focal Press ISBN 978-0-240-51540-3
35. Schwarting M, Burton T, Yampolskiy R (2016) On the obfuscation of image sensor fingerprints, 2015 annual global online conference on information and computer technology IEEE
36. Taspinar S, Mohanty M, Memon ND (2016) PRNU Based source attribution with a collection of seam-carved images. In: 2016 IEEE International Conference on Image Processing, ICIP, pp 156–160, vol 2016
37. Yao H, Wang S, Zhang X, Qin C, Wang J (2017) Detecting image splicing based on noise level inconsistency. Multimed Tools Appl 76:12457–12479
38. Yao H, Cao F, Tang Z, Wang J, Qiao T (2018) Expose noise level inconsistency incorporating the inhomogeneity scoring strategy. Multimed Tools Appl 77:18139–18161
39. Zheng Y, Lin S, Kambhamettu C, Yu J, Kang SsB (2009) Single-Image Vignetting correction. IEEE Trans Pattern Anal Mach Intell 31(12):2243–2256
40. Zheng Y, Lin S, Kang SB, Xiao R, Gee JC, Kambhamettu C (2013) Single-Image Vignetting correction from gradient distribution symmetries. IEEE Trans Pattern Anal Mach Intell 35(6):1480–1494
41. Zeng H, Chen J, Kang X, Zeng W (2015) Removing camera fingerprint to disguise photograph source. In: 2015 IEEE International Conference on Image Processing, ICIP 2015, Quebec City, QC, Canada, September 27–30, 2015, IEEE, pp 1687–1691
42. Zeng H, Liu J, Yu J, Kang S (2017) A framework of camera source identification bayesian game. IEEE Transactions on Cybernetics 47(7):1757–1768



Jarosław Bernacki born in 1989. Ph. D. candidate in Wrocław University of Science and Technology, Poland. His main research interests include image processing, anonymity, privacy, cryptography and computer networks.