ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10096605

EXPLOITING PRNU AND LINEAR PATTERNS IN FORENSIC CAMERA ATTRIBUTION UNDER COMPLEX LENS DISTORTION CORRECTION

Andrea Montibeller[⋆] *Fernando Pérez-González* [†]

[⋆]University of Trento, Department of Information Engineering and Computer Science, Italy
[†]atlanTTic, University of Vigo, Department of Signal Theory and Communications, Spain

ABSTRACT

More complex and ever more common lens distortion correction post-processing is seriously hampering state-of-the-art camera attribution techniques. In this paper, we show that the two main existing techniques, namely PRNU (Photo Response Non Uniformity)-based and linear-pattern-based, can be successfully combined to improve performance. Moreover, we introduce a novel method that is able to correctly invert adaptive distortion correction transformations by successively maximizing the peak-to-correlation energy (PCE) and the linear-pattern energy for much more reliable camera attribution. A novel validation procedure to quickly discard mismatched test images is also proposed. Finally, we show how great reductions in running time can be achieved by using a GPU for interpolation, resampling, and PCE computation. The code is available at `https://github.com/AMontiB/PSLR`.

***Index Terms*—** PRNU, Camera Attribution, Camera Source Identification, Radial Distortion Correction, GPU

1. INTRODUCTION

Camera attribution is the task of determining whether a given device (i.e. smartphone or digital camera) was used to take a given image. The last two decades have witnessed a number of solutions, some of them aimed at identifying just the model of the device by studying features introduced during the image acquisition process [1][2][3][4], and others the device, by exploiting invisible residuals introduced by the sensor of the camera [5, 6]. Of all those features, the most accurate for camera attribution purposes, remains the Photo Response non-Uniformity (PRNU) [5].

The PRNU is a unique noise-like weak signal due to manufacturing imperfections of the sensor which appears superimposed on the captured image [5]. A device's PRNU can be extracted from images taken with it and compared with a residual obtained from an image of the same or different devices. The PRNU is very sensitive to image post-processing, that hampers its reliability [7]. Specifically, when spatial transformations such as cropping, up/down-sampling [8], radial distortion correction [9][10], HDR post-processing [11], video stabilization [12], and others, [7] are applied to the host image, the PRNU also gets distorted and becomes unreliable unless the spatial transformation is properly inverted.

In this paper we will focus on in-camera and out-camera radial corrections and on the effects they have on the PRNU. In-camera radial corrections are common post-processing techniques used in modern devices to mitigate simple barrel or pincushion distortions due to camera lenses [13]. In the last few years, more complex and adaptive radial corrections, able to fit almost perfectly the lens distortion model, have become available. Those complex corrections are handled out-camera by third-party editing software like Adobe Lightroom, Pt-Lens, Photoshop, and Gimp, seriously impairing the performance of state-of-the-art methods [9], [10] which were mostly developed to invert simple barrel and pincushion radial corrections.

This work analyzes the main limitations affecting [9], [10] when applied to modern, more sophisticated radial corrections. Furthermore, we will present novel solutions to improve the accuracy of [9] and [10]: first, we will show the effectiveness of a proper combination of both methods; second, we propose a coarse parameter estimation guided by the Peak-to-Correlation Energy (PCE) followed by a refinement based on the Linear Pattern Energy (LPE). Moreover, we propose a novel validation procedure based on the PCE that serves to quickly discard test images without having to wait for the entire hypothesis test in order to declare a mismatch. Finally, similarly to [14], we show how the required resampling, interpolation, and PCE computing operations can be sped-up by running them on a GPU.

The paper is organized as follows: in Section 2 we will present the background, including the state of the art and its limitations; in Section 3 we discuss our contributions. Section 4 empirically validates our proposals, and, finally, Section 5 provides some conclusions.

2. BACKGROUND

In this paper, images are represented by matrices of size $M \times N$ and denoted in boldface, e.g. $\mathbf{X}$. The normalized cross-correlation (NCC) between $\mathbf{X}$ and $\mathbf{Y}$ is denoted by $\rho(\mathbf{X}, \mathbf{Y})$ and is defined as the NCC between the column vectors that are obtained by stacking the columns of $\mathbf{X}$ and $\mathbf{Y}$; if $\mathbf{X}$ and

$\mathbf{Y}$ have different sizes, we centrally crop the bigger matrix to fit the size of the smaller one, similarly to [10]. Given a set $\mathcal{S}$, its cardinality is denoted by $|\mathcal{S}|$. Given an image $\mathbf{X}$ and a vector $\boldsymbol{\delta} = (\delta_1, \delta_2) \in \mathbb{Z}^2$, $C(\mathbf{X}, \boldsymbol{\delta})$ denotes the image after a horizontal (vertical) cyclic shift of δ_1 (resp. δ_2) pixels.

2.1. PRNU

The PRNU is a weak, multiplicative, noise-like signal introduced by the camera sensor on every image it acquires. The uniqueness of the PRNU, due to specific manufacturing conditions [5], is perfectly suited to camera attribution. PRNU-based attribution requires a pre-processing that removes other components [5]. Typically, this task consists in applying a denoiser (customarily [15], as in this paper) $F(\cdot)$ to an image $\mathbf{I}$ to obtain a residual $\mathbf{W} \doteq \mathbf{I} - F(\mathbf{I})$ containing the PRNU.

To solve the attribution problem, we assume the existence of an estimate $\hat{\mathbf{K}}$ of the true PRNU $\mathbf{K}$ for the test camera, and the residual $\mathbf{W}$ from the test image $\mathbf{I}$. Given L images $\mathbf{I}_l$ with residuals $\mathbf{W}_l$, $l = 1, \cdots, L$, the PRNU can be estimated as [16]:

$$\hat{\mathbf{K}} = \left(\sum\nolimits_{l=1}^{L} \mathbf{I}_l \circ \mathbf{W}_l \right) \oslash \left(\sum\nolimits_{l=1}^{L} \mathbf{I}_l \circ \mathbf{I}_l \right), \qquad (1)$$

where $\circ$ and $\oslash$ denote element-wise product and division, respectively. The attribution test is a binary hypothesis test in which H_1 corresponds to the test image $\mathbf{I}$ containing the PRNU $\mathbf{K}$ of the test camera; else, H_0 is decided [17]. In [2] the PCE test statistic proposed in [16] is improved by considering the sign of the normalized cross-correlation; the resulting test statistic is the signed-PCE (sPCE), which in the case where $\mathbf{I}$ and $\hat{\mathbf{K}}$ are aligned takes the following form:

$$\text{sPCE}(\hat{\mathbf{K}}, \mathbf{I}) \doteq \frac{\text{sgn}(\rho(\hat{\mathbf{K}}, \mathbf{W})) \cdot \rho^2(\hat{\mathbf{K}}, \mathbf{W})}{\frac{1}{MN - |\mathcal{D}|} \sum_{\boldsymbol{\delta} \in \mathcal{I} \setminus \mathcal{D}} \rho^2(\hat{\mathbf{K}}, C(\mathbf{W}, \boldsymbol{\delta}))}, \qquad (2)$$

where $\mathcal{I}$ is the set of image pixel-coordinates and $\mathcal{D}$ is a cyclic exclusion neighborhood around the origin (in this paper, of size 11×11 pixels) to avoid contamination of cross-correlation peaks from H_1 when estimating the cross-correlation noise under H_0 [8].

2.2. Radial Correction

The radial corrections analyzed in this paper are radially-symmetric [9] barrel/pincushion distortions applied in-camera by compact devices, and adaptive ones applied out-camera by editing software like Adobe Lightroom which exploit more powerful hardware and databases of camera models.

According to [10], [18], if (x', y') are the image coordinates after radial distortion correction, there exists a geometrical mapping $G_{\boldsymbol{\alpha}}$ such that $G_{\boldsymbol{\alpha}} : \mathbb{R}^2 \to \mathbb{R}^2$, and $(x', y') \to (x, y)$, that inverts the correction, and that can be written as:

$$(x, y) = (x'_0, y'_0) + \left[(x', y') - (x'_0, y'_0) \right] \left(1 + \sum\nolimits_{i=1}^{n} \alpha_i r'^{2i} \right) \qquad (3)$$

where (x'_0, y'_0) are the optical center coordinates, $\boldsymbol{\alpha} \doteq [\alpha_1, \cdots, \alpha_n]^T$ contains the parameters of the mapping, and $r'^2 = 4 \cdot [(x' - x'_0)^2 + (y' - y'_0)^2]/D^2$ is the normalized squared distance from the point (x', y') so that $r' = 1$ corresponds to half of the image diagonal D. By assuming that $G_{\boldsymbol{\alpha}}(x'_0, y'_0) = (x'_0, y'_0)$, with a slight abuse of notation, and dropping the phase component, (3) can be rewritten in normalized radial coordinates as:

$$r = G_{\boldsymbol{\alpha}}(r') = r' \left(1 + \sum\nolimits_{i=1}^{n} \alpha_i r'^{2i} \right) \qquad (4)$$

In this paper, $n = 2$, i.e. we will consider distortion correction models with two parameters, α_1 and α_2.

2.3. Camera Attribution of Radial Corrected Images

The best performing approaches for camera attribution of radially corrected images were proposed by Goljan et. al. in [9] and [10], and rely on two different strategies to estimate the parameter vector $\boldsymbol{\alpha}$ such that $G_{\boldsymbol{\alpha}}$ best inverts the radial correction distortion. The difference between those two methods lies in their objective functions, as we discuss next.

1) *PCE-guided method* [9]: It uses a combination of a grid-search and a golden-search to estimate $\boldsymbol{\alpha} = [\alpha_1, \alpha_2]^T$, with the additional constraint $\alpha_2 = -3\alpha_1^2$, aiming at maximizing the PCE. During the grid search, a coarse estimation of α_1 is obtained and further refined through four iterations, each time with a smaller stepsize.

This works well on high-resolution in-camera-corrected images but its performance dramatically decreases under complex and adaptive radial corrections like those of Lightroom, and for low-resolution images. The issue with complex corrections is due to the the above constraint, which can be seen that corresponds to the inverse transformation of simple barrel/pincushion mappings. The drop for low-resolution images owes to downsampling (by a factor of 2×2) of image residuals. Such a downsampling is needed to deal with the high computational complexity of this method.

2) *LPE-guided method* [10]: It uses the linear pattern energy (LPE) as the objective function to guide the estimation of $\boldsymbol{\alpha} = [\alpha_1, \alpha_2]^T$. Those linear grid-like patterns are camera artifacts that show up in the residuals of non-corrected images; since distortion correction transforms these patterns, the correct inverse mapping $G_{\boldsymbol{\alpha}}$ would be such that when it is applied they are restored. Parameter estimation proceeds then in two stages: in the first, a grid search is employed to locate that value of α_1 that maximizes the LPE. In the second stage, the algorithm refines this value of α_1 and estimates α_2 by using the Nelder-Mead method. The LPE has a bias that depends on $\boldsymbol{\alpha}$, and is quite noisy; to solve these issues, the algorithm not only removes the bias by computing its trend through second-order polynomial fitting, but it also validates the peak by comparing it with a threshold.

Similarly to [9], the method in [10] performs well on high-resolution in-camera-corrected images, but behaves poorly on

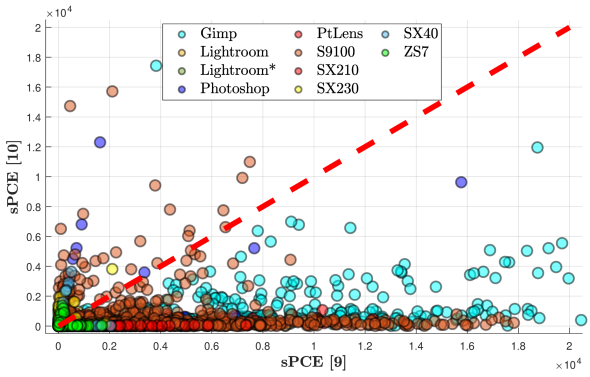

Fig. 1: PCE values obtained with [9] and [10] on the test dataset. Samples near the dashed line corresponds to similar output PCE values.

images either with low-resolution or corrected out-camera. This can be attributed to [10] only considering barrel-like mappings and to its validation procedure not taking advantage of the PCE for a further check.

3. PROPOSED SOLUTIONS

During our experiments, we noticed that *PCE-based* [9] and *LPE-based* [10] methods performed differently depending on the type of radial corrections. This is illustrated in Fig. 1, where the sPCE values of [9] and [10] are plotted against each other for different subsets of our test database; the diagonal corresponds to equal performance. These large deviations are due to different intrinsic features and limitations of both methods: the one in [9] works well when it comes to inverting both barrel and pincushion radial corrections, but struggles when trying to invert complex radial corrections that employ more than one parameter. In the latter case, [10] performs generally better, especially when $\alpha_1 \in [0, 0.33]$.

This observation is key to proposing combinations of [9] and [10] using some judicious criterion; here we will discuss MAX and OR rules. The MAX rule consists in running the two methods in parallel (MAXpar) or in sequence (MAXseq), on the same image, taking the maximum of both, and comparing it with a threshold experimentally set to achieve a False Positive Rate (FPR) ≈ 0.05. In this way, we choose the parameter vector $\boldsymbol{\alpha}$ that maximizes the final PCE value. In contrast, the OR criterion declares a match whenever either method yields a PCE value larger than its respective threshold. Both OR and MAX rules outperform [9] and [10] on low-resolution images, like those from the Panasonic ZS7 device, and complex out-camera radial corrections like those of Adobe Lightroom, see Sec. 4. This further illustrates some complementarity (to the best of our knowledge up to now untapped) of [9] and [10], which can also be appreciated in Fig. 1. Unfortunately, both OR and MAX criteria increase the computational complexity of the original methods (already computationally quite demanding); in addition, they do not fully complement each other towards an accurate estimation of the spatial transformation parameters α_1 and α_2.

Alternatively, we propose a PCE-guided coarse parameter Search plus LPE-based Refinement (PSLR), that more efficiently leverages the advantages of [9] and [10]. This approach first estimates α_1, when $\alpha_2 = 0$, using the grid search of [9] but without its early stopping conditions, which is more accurate than the one of [10] and is able to cover both positive and negative values of α_1. Let $\hat{\alpha}_1$ denote the estimate just described. Towards a fast detection of mismatches, the algorithm verifies the correctness of $\hat{\alpha}_1$; to that end, we do not follow the same validation procedure of [10], but we evaluate the PCE peak obtained at $\hat{\alpha}_1$ by checking that:

$$\text{sPCE}(\hat{\mathbf{K}}, G_{(\hat{\alpha}_1,0)}(\mathbf{W})) - \mu > \tau_v \tag{5}$$

where

$$\mu \doteq \Big(\text{sPCE}(\hat{\mathbf{K}}, G_{(\hat{\alpha}_1+\Delta,0)}(\mathbf{W})) + \text{sPCE}(\hat{\mathbf{K}}, G_{(\hat{\alpha}_1-\Delta,0)}(\mathbf{W}))\Big)/2$$

for some values of Δ and τ_v to be discussed next. The intuition behind the proposed validation is that the sPCE exhibits a sharp peak if represented against α_1 (when $\alpha_2 = 0$); to validate whether $\hat{\alpha}_1$ corresponds to the peak, we subtract from the sPCE at $\hat{\alpha}_1$ the average sPCE obtained at two points away from the presumed peak, and compare the result with a threshold. This threshold has been experimentally set to achieve a probability of false validation of 0.01 (under both H_0 and H_1 hypotheses) using 200 images from our database. Following the previous discussion, we set $\Delta = 0.1$ and $\tau_v = 10.58$.

If (5) is satisfied, the PSLR algorithm proceeds with the estimation of α_2; otherwise, a mismatch is declared. To estimate α_2, we maximize the LPE using as in [10] the Nelder-Mead algorithm which demonstrated, during our experiments, to be very robust and reliable if initialized using the correct value of α_1. The algorithm decides H_1 if, during any of its two stages, an sPCE larger than τ is retrieved, where τ is a threshold set experimentally to achieve FPR ≈ 0.05.

To further reduce the computational cost of the PSLR algorithm, all the radial correction operations, interpolation, resampling, and PCE estimation are run on a GPU, as it is done in [14]. By doing so, we can parallelize all those operations that are most time-consuming when run on a CPU. However, for a fair comparison with the state of the art, we will also show the CPU run-times of PSLR. We note that PSLR uses the full-resolution noise residual $\mathbf{W}$ and not its downsampled version as in [9].

4. EXPERIMENTAL RESULTS

We compared our solutions with the PCE-based method of [9], also without downsampling (noDS), and the LPE-based method of [10] (all of the previous using the sPCE for improved performance) in terms of True Positive Rate (TPR) for an FPR ≈ 0.05, Area Under the Curve for FPR < 0.05 (AUC@0.05, which is normalized to yield 1 in the ideal

	GIMP 3456 × 5184		LIGHTROOM 3456 × 5184		LIGHTROOM* 3456 × 5184		PHOTOSHOP 3456 × 5184		PT LENS 3456 × 5184		S9100 3000 × 4000		SX210 3240 × 4320		SX230 1584 × 2816		SX40 2664 × 4000		ZS7 1920 × 2560	
	TPR	time [s]	TPR	time [s]	TPR	time [s]	TPR	time [s]	TPR	time [s]	TPR	time [s]	TPR	time [s]	TPR	time [s]	TPR	time [s]	TPR	time [s]
[9] ($\tau = 4.81$)	0.96	85.3	0.44	87.9	0.41	91.6	0.91	107.2	0.64	100.3	0.97	81.7	0.98	81.3	0.82	25.1	0.98	56.3	0.79	27.8
[9] no DS ($\tau = 7.65$)	0.97	962.3	0.6	930.8	0.51	930.9	**0.96**	947.6	**0.91**	918.3	**1**	598.3	**1**	723.4	0.98	229.0	**1**	526.3	**0.98**	255.2
[10] ($\tau = 2.83$)	0.96	861.7	0.35	849.8	0.55	808.7	0.88	731.7	0.36	847.1	0.92	553.5	0.93	661.1	0.76	205.2	0.70	472.9	0.65	227.2
MAXpar ($\tau = 5.28$)	0.97	861.7	0.68	849.8	**0.75**	808.7	0.93	731.7	0.83	847.1	**0.99**	553.5	**1**	661.1	0.88	205.2	**1**	472.9	0.87	227.2
MAXseq ($\tau = 5.28$)	0.97	947	0.68	937.3	**0.75**	900.3	0.93	838.9	0.83	947.4	**0.99**	635.2	**1**	742.4	0.88	230.3	**1**	529.2	0.87	255
OR ($\tau = 5.28$)	0.96	96.4	0.67	548.1	0.74	553.8	0.93	171.6	0.83	402.4	**0.99**	95.8	**1**	93.2	0.88	60.3	**1**	67.2	0.86	70.1
PSLR ($\tau = 5.83$)	**0.98**	197.1	**0.75**	308.6	0.71	284.2	**0.96**	162.2	**0.9**	140.3	0.96	80.72	**1**	234.1	**0.98**	147.2	**1**	125.4	**0.95**	51.8
PSLR CPU ($\tau = 5.83$)	**0.98**	667.76	**0.75**	932.29	0.71	962.46	**0.96**	576.83	**0.9**	474.04	0.96	197.75	**1**	621.83	**0.98**	363.18	**1**	287.46	**0.95**	151.82

Table 1: TPR and average execution time of [9], [10] and our proposed schemes, for different subsets.

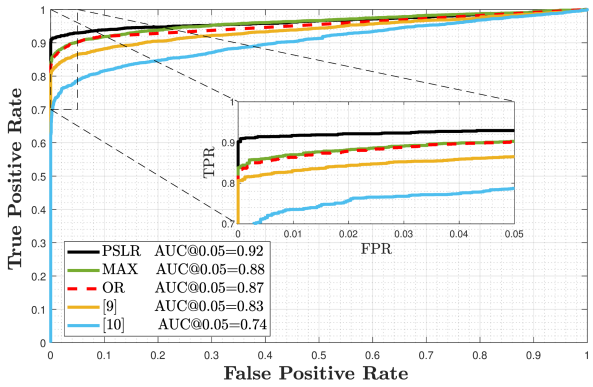

Fig. 2: ROCs obtained with [9], [10], OR, MAX and PSLR on the dataset of this paper.

case), Receiver Operating Characteristic (ROC) curve, and time consumption.

The dataset used for testing is composed of 3645 images, of which, 2037 were downloaded from flicker and radially corrected in-camera by Canon SX230 HS, Panasonic ZS7, Canon SX40, Canon SX210, and Nikon S9100 models, and 1508 were taken with the Canon 1200D and radially-corrected out-camera by editing programs such as Adobe Lightroom Classic CC 2017, Adobe Photoshop CC 2017, PT Lens v2.0 (Macbook) and Gimp 2.10.14 (377 images by each software). In addition, we radially corrected 100 images taken with the Canon 1200D with Lightroom using the radial distortion models of other lenses: Nikon (20 images), Tamron (20), Apple (20), Huawei (20) and DJI (20). We refer to this subset as LIGHTROOM* in Table 1.

Further details on the dataset and how the camera PRNUs were estimated are available in [19]. All the experiments were run on a server with the following characteristics: RAM 256GB, Processor Intel(R) Xeon(R) CPU E5-2640 v4 2.40GHz, GPU NVIDIA QUADRO 22 GB. The average memory required for an image of size 3456×5184 is $\approx$ 9GB of GPU and $\approx$ 4GB of CPU.

Fig. 2 shows the ROCs obtained by [9], [10], and our proposed schemes. Table 1 provides more details, including the per-correction-type average time required to declare a match.

The results we obtained using the MAX and OR implementations demonstrate both the complementarity of [9] and [10] and large improvements, especially in presence of the most difficult radial corrections of our dataset (i.e. Lightroom, Lightroom*, and Pt-Lens) and low-resolution images (i.e. sx230 and zs7). These improvements are even more evident with PSLR, which yields the best results in terms of ROC, AUC and, on average, in terms of TPR. Through bootstrapping [20], we tested the null hypothesis that the AUC@0.05 of PSLR is smaller than that of the other methods, and is rejected with the following p-values: MAX: 0.033; OR: 0.024; [9] and [10]: < 0.001. From Table 1 it is possible to notice that improvements are also achievable when [9] works with no DS. However, its computational cost is very high, requiring up to 16 minutes to process a single high-resolution image; PSLR, running on a GPU, is much faster. This observation further corroborates the suitability of the GPU in camera attribution problems involving complex spatial transformations. Finally, by using (5), we drastically reduced the number of images wrongly labeled as radially corrected while rapidly detecting mismatches.

5. CONCLUSIONS

Despite the ever more common presence of distortion-corrected images, due to the availability of more powerful in-camera firmware and out-camera software, little progress has been done after the seminal works in [9] and [10]. This is particularly striking because the corrections have become much more complex than for simple barrel and pincushion distortions, where those methods struggle (none of them were able to achieve TPR> 0.75 at FPR ≈ 0.05 on the most difficult radial corrections of our dataset, corresponding to Lightroom and PT Lens software). Considering the continuous evolution of post-processing software, this situation can only get worse if no new advances are made.

In this paper we have taken a step in this direction by combining and improving the available methods. In particular, we have shown that PSLR, which finds the radial transformation parameters following a two-stage procedure with an sPCE-guided coarse search and a linear pattern-based refinement, is able to overcome the limitations of the state-of-the-art.

ACKNOWLEDGMENTS

Partially funded by MCIN/AEI/10.13039/501100011033 and NextGenerationEU/PRTR under project FELDSPAR, and by Xunta de Galicia and ERDF under project ED431C 2021/47.

6. REFERENCES

[1] Alessandro Piva, "An overview on image forensics," *International Scholarly Research Notices*, vol. 2013, 2013.

[2] X. Kang, Y. Li, Z. Qu, and J. Huang, "Enhancing source camera identification performance with a camera reference phase sensor pattern noise," *IEEE TIFS*, vol. 7, no. 2, pp. 393–402, 2012.

[3] Kai San Choi, Edmund Y Lam, and Kenneth KY Wong, "Source camera identification using footprints from lens aberration," in *SPIE*, 2006, vol. 6069, pp. 172–179.

[4] D. Cozzolino and L. Verdoliva, "Noiseprint: A cnn-based camera model fingerprint," *IEEE TIFS*, vol. 15, 2020.

[5] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE TIFS*, vol. 1, no. 2, pp. 205–214, 2006.

[6] A. E. Dirik, H. T. Sencar, and N. Memon, "Source camera identification based on sensor dust characteristics," in *2007 IEEE Workshop on Signal Processing Applications for Public Security and Forensics*. IEEE, 2007, pp. 1–6.

[7] Massimo Iuliani, Marco Fontani, and Alessandro Piva, "A leak in PRNU based source identification—questioning fingerprint uniqueness," *IEEE Access*, vol. 9, pp. 52455–52463, 2021.

[8] M. Goljan, "Digital camera identification from images–estimating false acceptance probability," in *International workshop on digital watermarking*. Springer, 2008, pp. 454–468.

[9] M. Goljan and J. Fridrich, "Sensor-fingerprint based identification of images corrected for lens distortion," in *Media Watermarking, Security, and Forensics 2012*. International Society for Optics and Photonics, 2012, vol. 8303, p. 83030H.

[10] M. Goljan and J. Fridrich, "Estimation of lens distortion correction from single images," in *Media Watermarking, Security, and Forensics 2014*. International Society for Optics and Photonics, 2014, vol. 9028, p. 90280N.

[11] M. Darvish Morshedi Hosseini and M. Goljan, "Camera identification from hdr images," in *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, 2019.

[12] Sara Mandelli, Paolo Bestagini, Luisa Verdoliva, and Stefano Tubaro, "Facing device attribution problem for stabilized video sequences," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 14–27, 2019.

[13] Seok-Han Lee, Sang-Keun Lee, and Jong-Soo Choi, "Correction of radial distortion using a planar checkerboard pattern and its image," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 1, pp. 27–33, 2009.

[14] Andrea Montibeller, Cecilia Pasquini, Giulia Boato, Stefano Dell'Anna, and Fernando Pérez-González, "GPU-accelerated sift-aided source identification of stabilized videos," in *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 2616–2620.

[15] M. K. Mihcak, I. Kozintsev, K. Ramchandran, and P. Moulin, "Low-complexity image denoising based on statistical modeling of wavelet coefficients," *IEEE Signal Processing Letters (SPL)*, vol. 6, pp. 300–303, 1999.

[16] M. Chen, J. Fridrich, M. Goljan, and J. Lukás, "Determining image origin and integrity using sensor noise," *IEEE TIFS*, vol. 3, no. 1, pp. 74–90, 2008.

[17] M. Goljan, J. Fridrich, and T. Filler, "Large scale test of sensor fingerprint camera identification," *Proceedings of SPIE - The International Society for Optical Engineering*, February 2009.

[18] Wolfgang Hugemann, "Correcting lens distortions in digital photographs," *Ingenieurbüro Morawski+ Hugemann: Leverkusen, Germany*, vol. 20, 2010.

[19] Andrea Montibeller and Fernando Pérez-González, "Technical report additional material for "an adaptive method for camera attribution under complex radial distortion corrections"," 2022, `http://dx.doi.org/10.13140/RG.2.2.20038.96323`.

[20] Patrice Bertail, Stéphan Clémençcon, and Nicolas Vayatis, "On bootstrapping the ROC curve," *Advances in Neural Information Processing Systems*, vol. 21, 2008.