An Adaptive Method for Camera Attribution Under Complex Radial Distortion Corrections

Andrea Montibeller, *Graduate Student Member, IEEE*, and Fernando Pérez-González, *Fellow, IEEE*

***Abstract*— Radial distortion correction, applied by in-camera or out-camera software/firmware alters the supporting grid of the image so as to hamper PRNU-based camera attribution. Existing solutions to deal with this problem try to invert/estimate the correction using radial transformations parameterized with few variables in order to restrain the computational load; however, with ever more prevalent complex distortion corrections their performance is unsatisfactory. In this paper we propose an adaptive algorithm that by dividing the image into concentric annuli is able to deal with sophisticated corrections like those applied out-camera by third party software like Adobe Lightroom, Photoshop, Gimp and PT-Lens. We also introduce a statistic called cumulative peak of correlation energy (CPCE) that allows for an efficient early stopping strategy. Experiments on a large dataset of in-camera and out-camera radially corrected images and on a in-the-wild dataset of images from smartphones show that our solution improves the state of the art in terms of both accuracy and computational cost.**

***Index Terms*— Image forensics, source attribution, PRNU, photo response non-uniformity, radial correction, distortion correction, PCE, adaptive processing.**

I. Introduction

DURING the past years, camera fingerprints based on the Photo Response Non-Uniformity (PRNU) have gained broad popularity in forensic applications thanks to their ability to identify the device that captured a certain image. The PRNU is a multiplicative spatial pattern that owes its uniqueness to manufacturing imperfections that cause sensor elements to have minute area and substrate material differences that make them capture different amounts of energy even under a perfectly uniform flat field [1]. Applications of the PRNU in multimedia forensics go beyond camera identification from images [2] or videos [3], as they have also been used in detecting inconsistencies that reflect image manipulations [4].

Unfortunately, the fact that the PRNU can be accurately modeled as a white random process explains its sensitivity to geometric transformations that alter the image coordinates. Unless those transformations are reverted, standard detection statistics will perform poorly as they are roughly based on cross-correlations that yield very small values under grid misalignment. In the literature several methods have been proposed to deal with those spatial transformations, including digital zoom [5], video stabilization [6], high dynamic range (HDR) processing [7], and radial distortion corrections [8], [9]. It is in the context of the latter that we have developed the methodology presented in this work.

Radial distortion correction aims at digitally removing the distortion introduced by the camera lens. This kind of processing is becoming more pervasive as devices increase their computing capabilities; in-camera correction is now common in compact models, tablets and smartphones. On the other hand, out-camera corrections can be performed with powerful software like Adobe Lightroom, which are able to invert distortions almost perfectly by matching the model of the lens mounted on the camera. This is not done by applying conventional radial distortion models such as *barrel* or *pincushion* but by making use of complex models (i.e., with a large number of parameters). As a consequence, existing methods [8], [9] relying on models with at most two parameters will only partially succeed in dealing with camera attribution under these complex out-camera processing. Increasing the number of model parameters often constitutes an undesirable path because reverting the distortion corrections entails a grid search whose computational load grows exponentially with the number of unknowns.

In this work we propose a novel approach to PRNU-based camera attribution under radial distortion corrections that is able to deal with complex models without significantly increasing the computational burden. The main idea is to divide the image under test and the PRNU into a series of concentric annuli that are thin enough to be locally describable with a simple (i.e., linear or cubic) distortion model which allows for an equally simple inverse transformation. The annuli are traversed sequentially by keeping track of the *cumulative peak-to-correlation energy ratio* (CPCE), which is a statistic

Manuscript received 28 July 2022; revised 27 June 2023; accepted 31 August 2023. Date of publication 27 September 2023; date of current version 22 November 2023. This work supported in part by the Defense Advanced Research Projects Agency (DARPA) under Grant HR00112090136; in part by the PREMIER Project funded by the Italian Ministry of Education, University, and Research (MIUR); in part by the Spanish Ministry for Science and Innovation; in part by the Spanish Cybersecurity Institute (INCIBE) and NextGenerationEU/PRTR through Projects "Federated Learning With Model Ownership Protection and Privacy Armoring" under Grant MCIN/AEI/10.13039/501100011033; in part by the "TRUFFLES: Trusted Framework for Federated Learning Systems;" in part by the Spanish Ministry of Universities through a "Salvador de Madariaga" Grant; and in part by Xunta de Galicia and the European Regional Development Fund under Project ED431C 2021/47. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Tiziano Bianchi. *(Corresponding author: Fernando Pérez-González.)*

Andrea Montibeller is with the Department of Information Engineering and Computer Science, University of Trento, 38123 Trento, Italy (e-mail: andrea.montibeller@unitn.it).

Fernando Pérez-González is with the atlanTTic Research Center, Department of Signal Theory and Communications, Universidade de Vigo, 36310 Vigo, Spain (e-mail: fperez@gts.uvigo.es).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIFS.2023.3318933, provided by the authors.

Digital Object Identifier 10.1109/TIFS.2023.3318933

introduced in this work and used to decide whether the radially corrected test image contains the reference PRNU. In fact, the sequential nature of the procedure makes it possible to implement an early stopping strategy to declare a match without having to process all the annuli and thus saving computational time. Another key feature of our method is *adaptivity*: instead of carrying out a wide-interval search for the distortion parameters describing each annulus, an adaptive Least-Mean-Squares-like predictor updates the parameters of the previously processed annulus in order to narrow down the current parameter search. This leads to a large computational efficiency without giving up flexibility. In order to steer the search we propose and justify mathematically a new objective function.

Different variants of our method are evaluated in terms of accuracy and speed. We compare our method with the state of the art in [8] and [9] on a large dataset composed of images taken with: 1) compact devices and radially corrected in-camera, and 2) a reflex camera and radially corrected out-camera using different software tools. Our results show considerable performance improvements, especially on low-resolution images and in presence of complex radial distortion corrections. We also show an application of our method in discovering new device attribution matches that would otherwise be assigned to the null hypothesis in a database of smartphone images [10].

The rest of the paper is organized as follows: Sect. II provides the mathematical background and formulates the addressed problem. Sect. III discusses the relevant state of the art. Sect. IV is devoted to discussing the proposed method which in Sect. V is validated and compared with [8] and [9]. Finally, Sect.VI presents our conclusions.

II. Problem Formulation and Modeling

A. Notation

The images considered in this paper can be either gray-scale or RGB; however, since the processing for the latter is carried out separately for each color channel, for notational simplicity we discuss the case of mono-channel images and later explain how the three channels are combined in the case of RGB. Bi-dimensional signals will be denoted with boldface. For every such signal, a domain $\mathcal{S} \subset \mathbb{Z}^2$ will be specified; for instance, a signal $\mathbf{X}$ with domain $\mathcal{S}_X$ is a collection of values $X_{i,j} \in \mathbb{R}$ defined for all locations $(i, j) \in \mathcal{S}_X$. For the case of images of size $M \times N$, the original domain is $\mathcal{I} = \{1, \cdots, M\} \times \{1, \cdots, N\} \subset \mathbb{Z}^2$; however, we will often find ourselves working with domains that are subsets of $\mathcal{I}$. We will denote by D_2 half of the diagonal of domain $\mathcal{I}$ measured in pixels. Notice that the set $\mathcal{I}$ can be expressed as $\mathcal{I} = \mathcal{B} \cap \mathbb{Z}^2$, with $\mathcal{B} \subset \mathbb{R}^2$ denoting the image bounding box.

The inner product of two signals $\mathbf{X}$ and $\mathbf{Y}$ with respective domains $\mathcal{S}_X$ and $\mathcal{S}_Y$ can be defined by extending the Frobenius product of matrices as $\langle \mathbf{X}, \mathbf{Y} \rangle \doteq \sum_{(i,j)\in\mathcal{S}} X_{i,j} Y_{i,j}$, where $\mathcal{S} = \mathcal{S}_X \cap \mathcal{S}_Y$ is assumed to be non-empty. The Frobenius norm of $\mathbf{X}$ with domain $\mathcal{S}_X$ induced by this inner product is $||\mathbf{X}|| \doteq \langle \mathbf{X}, \mathbf{X} \rangle = \sum_{(i,j)\in\mathcal{S}_X} X_{i,j}^2$. The product of signals $\mathbf{X}$ and $\mathbf{Y}$, denoted by $\mathbf{X} \circ \mathbf{Y}$, is the element-wise product, i.e., $(\mathbf{X} \circ \mathbf{Y})_{i,j} = X_{i,j} \cdot Y_{i,j}$ and is defined for all $(i, j) \in \mathcal{S}_X \cap \mathcal{S}_Y$. The multiplicative inverse of $\mathbf{X}$ is denoted by $\mathbf{X}^{\circ -1}$ and is such that $(\mathbf{X}^{\circ -1})_{i,j} = X_{i,j}^{-1}$. For a signal $\mathbf{X}$ with domain $\mathcal{S}_X$, we denote by $\bar{\mathbf{X}}$ a constant signal with the same support as $\mathbf{X}$ and whose value is the sample mean $\sum_{(i,j)\in\mathcal{S}_X} X_{i,j}/|\mathcal{S}_X|$, where $|\mathcal{S}_X|$ denotes the cardinality of $\mathcal{S}_X$. The normalized cross-correlation (NCC) between $\mathbf{X}$ and $\mathbf{Y}$ is defined as

$$\rho(\mathbf{X}, \mathbf{Y}) = \frac{\langle \mathbf{X} - \bar{\mathbf{X}}, \mathbf{Y} - \bar{\mathbf{Y}} \rangle}{||\mathbf{X} - \bar{\mathbf{X}}|| \cdot ||\mathbf{Y} - \bar{\mathbf{Y}}||}, \tag{1}$$

with the inner product and norms defined as above.

Given a signal $\mathbf{X}$ with rectangular domain $\mathcal{I}$ and a vector $\mathbf{s} = (s_1, s_2) \in \mathbb{Z}^2$, we denote by $C(\mathbf{X}, \mathbf{s})$ the cyclic shift of $\mathbf{X}$ by vector $\mathbf{s}$, so that the (i, j)th component of $C(\mathbf{X}, \mathbf{s})$ is $X_{(i+s_1) \bmod M, (j+s_2) \bmod N}$. Note that the domain of $C(\mathbf{X}, \mathbf{s})$ is also $\mathcal{I}$. Finally, the all-zeros image is denoted by $\mathbf{0}$.

B. PRNU Estimation

As previously indicated, the PRNU is a multiplicative noise-like signal that serves as a sensor fingerprint [1], [5]. Because the PRNU is a very weak signal, it is necessary to separate it from both the true image and other noise components. If $\mathbf{I}_0$ denotes the image in absence of noise, and $\mathbf{K}$ is the PRNU, it is possible to derive the following simplified model [11]:

$$\mathbf{I} = \mathbf{I}_0 + \mathbf{I}_0 \circ \mathbf{K} + \Theta, \tag{2}$$

where Θ is uncorrelated with both $\mathbf{I}_0$ and $\mathbf{K}$, and summarizes noise components of different nature, and all signals are defined over $\mathcal{I}$. The fingerprint $\mathbf{K}$ of a camera can be extracted from L images $\mathbf{I}^{(l)}$, $l = 1, \cdots, L$, taken with the camera under analysis. Let $\mathbf{W}^{(l)}$ denote the noise *residual* obtained by applying a generic denoising filter $F(\cdot)$ to the lth image $\mathbf{I}^{(l)}$, as

$$\mathbf{W}^{(l)} = \mathbf{I}^{(l)} - F(\mathbf{I}^{(l)}), \quad l = 1, \cdots, L. \tag{3}$$

For RGB images, (3) is applied to each image color channel and the results converted into gray-scale using the standard linear combination [12]. In this case, $\mathbf{W}^{(l)}$ will denote the resulting combined residual. All our reported experiments have been carried out on color images, using Mihcak's wavelet-based denoiser [13] for it yields an excellent trade-off between performance and complexity. Then, the PRNU can be estimated as follows [11]:

$$\hat{\mathbf{K}} = \left(\sum_{l=1}^{L} \mathbf{I}^{(l)} \circ \mathbf{W}^{(l)} \right) \circ \left(\sum_{l=1}^{L} \mathbf{I}^{(l)} \circ \mathbf{I}^{(l)} \right)^{\circ -1}. \tag{4}$$

The estimate so obtained is customarily post-processed to remove some systematic artifacts that are present in most cameras. Here, we will follow [11] and apply a mean-removal operation by columns and rows, and a Wiener filter in the DFT aimed at removing periodic spatial artifacts. Similarly to (3), for color images the fingerprints are estimated separately for each channel and then linearly combined into gray-scale [12]; in this case, $\hat{\mathbf{K}}$ denotes the result of this conversion. Given an image under investigation $\mathbf{I}$ and its corresponding residual

$\mathbf{W} \doteq \mathbf{I} - F(\mathbf{I})$ a binary hypothesis test can be formulated to decide whether $\mathbf{I}$ contains a certain PRNU $\mathbf{K}'$ for which an estimate $\hat{\mathbf{K}}'$ is available. We will denote the null hypothesis of this test (i.e., $\mathbf{I}$ does not contain $\mathbf{K}'$) by H_0 and the alternative (i.e., $\mathbf{I}$ contains $\mathbf{K}'$) by H_1.

The most popular decision statistic for the test is the *Peak-to-Correlation Energy ratio* (PCE) which computes the peak cross correlation between the test image residual $\mathbf{W}$ and the estimated PRNU $\hat{\mathbf{K}}'$ from the candidate camera, and normalizes it by an estimate of the correlation noise under H_0 [12].

For non-cropped images the PCE simplifies to

$$\mathrm{PCE}(\hat{\mathbf{K}}', \mathbf{W}) = \frac{\mathrm{sgn}(\rho(\hat{\mathbf{K}}', \mathbf{W})) \cdot \rho^2(\hat{\mathbf{K}}', \mathbf{W})}{\frac{1}{|\mathcal{I}\setminus\mathcal{S}|}\sum_{\mathbf{s}\in\mathcal{I}\setminus\mathcal{S}} \rho^2(\hat{\mathbf{K}}', C(\mathbf{W}, \mathbf{s}))}, \tag{5}$$

where, following the improvement proposed in [14], we have included the sign of the NCC to exclude negative values that would be never expected under H_1. In (5) $\mathcal{S}$ is a *cyclic exclusion neighborhood* of $(0, 0)$ of small size (e.g., 11×11 pixels) to avoid contamination from cross-correlation peaks when estimating the cross-correlation noise under H_1. Noticing that for every $\mathbf{s}$, $||C(\mathbf{W}, \mathbf{s}) - \overline{C(\mathbf{W}, \mathbf{s})}|| = ||\mathbf{W} - \bar{\mathbf{W}}||$, and letting $\tilde{\mathbf{W}} \doteq \mathbf{W} - \bar{\mathbf{W}}$, (5) can be alternatively written as

$$\mathrm{PCE}(\hat{\mathbf{K}}', \mathbf{W}) = \frac{\mathrm{ssq}(\langle\hat{\mathbf{K}}', \tilde{\mathbf{W}}\rangle)}{\frac{1}{|\mathcal{I}\setminus\mathcal{S}|}\sum_{\mathbf{s}\in\mathcal{I}\setminus\mathcal{S}} \langle\hat{\mathbf{K}}', C(\tilde{\mathbf{W}}, \mathbf{s})\rangle^2}, \tag{6}$$

where we have assumed that the mean of $\hat{\mathbf{K}}'$ is zero due to the zero-meaning operation discussed above, and the signed-squared function $\mathrm{ssq}(\cdot)$ is such that $\mathrm{ssq}(x) \doteq \mathrm{sgn}(x) \cdot x^2$.

C. Lens Distortion Models

To describe radially symmetric barrel/pincushion distortions we adopt the same models presented in [8] and explained in [15], [16], and [17]. If we denote the coordinates before and after the radial distortion by (x, y) and (x', y'), xrespectively, the invertible geometrical mapping T_α is given by

$$\begin{aligned} T_\alpha : \mathbb{R}^2 &\to \mathbb{R}^2 \\ (x, y) &\mapsto (x', y') \end{aligned} \tag{7}$$

where

$$x' = x_p + (x - x_p)(1 + \alpha r^2); \tag{8}$$

$$y' = y_p + (y - y_p)(1 + \alpha r^2), \tag{9}$$

and (x_p, y_p) is the *optical center* of the image and $r^2 \doteq [(x - x_p)^2 + (y - y_p)^2]/D_2^2$ is the normalized squared radial distance from point (x, y) to the optical center. This normalization by D_2^2 is for convenience, so that $r = 1$ corresponds to half of the image diagonal [8]. Parameter $\alpha \in \mathbb{R}$ in (8-9) models the type of radial distortion: $\alpha > 0$ for pincushion distortion, and $\alpha < 0$ for barrel distortion. Alternatively, given (x_p, y_p) and assuming that $T_\alpha(x_p, y_p) = (x_p, y_p)$, the transformation can be written in normalized polar coordinates. Since the phase is preserved under $T_\alpha(\cdot)$, with a slight abuse of notation we will drop the phase component and sometimes write the radial transformation as $T_\alpha : \mathbb{R}^+ \cup \{0\} \to \mathbb{R}^+ \cup \{0\}$ such that

$$r' = T_\alpha(r) = r(1 + \alpha r^2). \tag{10}$$

More complex radial corrections [9], [18] can be expressed through an nth order model:

$$r' = T_{\boldsymbol{\alpha}}(r) = r\left(1 + \sum_{i=1}^{n} \alpha_i r^{2i}\right), \tag{11}$$

where $\boldsymbol{\alpha} \doteq [\alpha_1, \cdots, \alpha_n]^T$ is a real parameter vector. In this paper, we consider that all radial corrections that cannot be expressed or well approximated with less than three non-null parameters are "complex". It is also important to remark that parameter vector $\boldsymbol{\alpha}$ often depends on the camera settings, such as the lens type or the focal distance, so images taken with the same camera may experience different distortion corrections.

Again, with some abuse of notation, and following [8], given a signal $\mathbf{X}$ with domain $\mathcal{S}_X$, the mapping $\mathbf{Y} = T_{\boldsymbol{\alpha}}(\mathbf{X})$ is produced as follows. Let $\mathbf{X}'$ be the signal with domain $\mathcal{S}_{X'} = T_{\boldsymbol{\alpha}}(\mathcal{S}_X)$ such that, for every $(u, v) \in \mathcal{S}_X$, and with $(u', v') = T_{\boldsymbol{\alpha}}(u, v)$, $X'_{u',v'} = X_{u,v}$. Then, given an ouput domain $\mathcal{S}_Y$, the signal $\mathbf{Y} = T_{\boldsymbol{\alpha}}(\mathbf{X})$ is obtained by interpolating the signal $\mathbf{X}'$ defined on $\mathcal{S}'_X$ at the points in $\mathcal{S}_Y$. Of course, precautions must be taken when specifying $\mathcal{S}_Y$ so that the interpolation is computable at all points in $\mathcal{S}_Y$. This aspect will be made clearer in Sect. IV, when we present our method.

D. Direct and Inverse Approaches to PCE Computation

When the image under analysis has been subjected to a radial distortion correction, the statistic $\mathrm{PCE}(\hat{\mathbf{K}}', \mathbf{W})$ is expected to perform poorly under H_1 in the hypothesis test, because the grids supporting $\hat{\mathbf{K}}'$ and $\mathbf{W}$ will not coincide (recall that the PRNU has a very narrow spatial autocorrelation function).

The approach explored in [8] is to take into account the distortion correction when computing the PCE. If the parameter vector $\boldsymbol{\alpha}$ of the radial mapping is known, there are essentially two possibilities, which we will term *direct* and *inverse*. In the direct approach, the candidate PRNU $\hat{\mathbf{K}}'$ is transformed in order for its grid to match that of $\mathbf{W}$. Then, the test statistic becomes

$$\mathrm{PCE}_{\mathsf{dir}}(\boldsymbol{\alpha}) \doteq \mathrm{PCE}(T_{\boldsymbol{\alpha}}(\hat{\mathbf{K}}'), \mathbf{W}), \tag{12}$$

where the domain $\mathcal{I}_T$ of $T_{\boldsymbol{\alpha}}(\hat{\mathbf{K}}')$ is the largest rectangular subset of $\mathcal{I}$ for which the interpolation is computable (see discussion at the end of Sect. II-C) and, accordingly, $\mathcal{I}_T$ replaces $\mathcal{I}$ in the denominator of (5).

In the inverse approach $\mathbf{W}$ is mapped back to the original domain, so that its grid coincides with that of $\hat{\mathbf{K}}'$. Then, the test statistic in this case is

$$\mathrm{PCE}_{\mathsf{inv}}(\boldsymbol{\alpha}) \doteq \mathrm{PCE}(\hat{\mathbf{K}}', T_{\boldsymbol{\alpha}}^{-1}(\mathbf{W})). \tag{13}$$

where, as above, the domain $\mathcal{I}_T$ of $T_{\boldsymbol{\alpha}}^{-1}(\mathbf{W})$ is the largest rectangular subset of $\mathcal{I}$ for which the interpolation is computable and $\mathcal{I}_T$ replaces $\mathcal{I}$ in the denominator of (6).

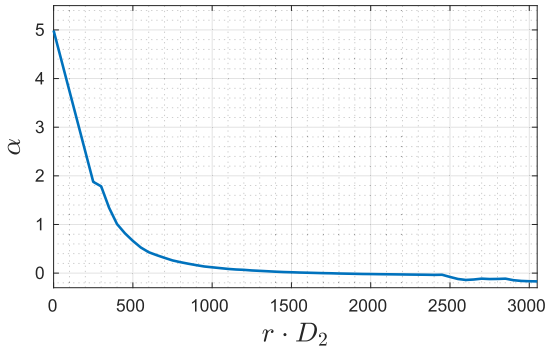

Fig. 1. Values of α maximizing $\text{PCE}_{\text{inv}}(\alpha)$ vs inner radius of the annulus. Values are linearly interpolated. Canon 1200D camera with EF-S 10-18mm lens, corrected with Adobe Lightroom. Focal length: 10mm. Shutter speed: 1/100 sec. Aperture: f7.1. ISO 800. The PRNU is estimated with 20 natural images all taken with those settings.

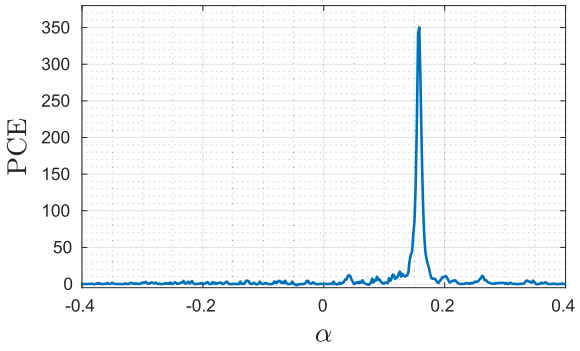

Fig. 2. PCE_{max} as a function of α for a Panasonic DMC-ZS7 camera.

Since one is interested in finding the best possible match, [8] suggests using the following statistic

$$\text{PCE}_{\text{max}}(\boldsymbol{\alpha}) \doteq \max\{\text{PCE}_{\text{dir}}(\boldsymbol{\alpha}), \text{PCE}_{\text{inv}}(\boldsymbol{\alpha})\}. \tag{14}$$

When the parameter vector $\boldsymbol{\alpha}$ is not known, which is often the case in practice, it must be estimated. In [8] this is done by maximizing the test statistic in (12-13), which makes sense from a maximum likelihood point of view. Let $\mathcal{A} \subset \mathbb{R}^n$ be the set of feasible vectors $\boldsymbol{\alpha}$; then, the statistic used in the hypothesis test is

$$\text{PCE}^*_{\text{max}} \doteq \max_{\boldsymbol{\alpha}\in\mathcal{A}} \text{PCE}_{\text{max}}(\boldsymbol{\alpha}). \tag{15}$$

For the case of scalar $\boldsymbol{\alpha}$ in (10) the inverse radial correction $T^{-1}_{\boldsymbol{\alpha}}(\mathbf{W})$ needed in (13) can be approximated via the Lagrange Inversion Theorem [19, 3.6.6.] which yields

$$r = T^{-1}_{\alpha}(r') = r'(1 - \alpha r'^2 + 3\alpha^2 r'^4 + O(r'^6)). \tag{16}$$

Using the approach described above, the radial correction can be approximately inverted in many practical cases by finding the optimal value of α [8]. However, when more complex radial corrections as in (11) have been applied, a single parameter α may be not sufficient. To illustrate this fact, we consider the example of an image of size 3456×5184 taken with a Canon 1200D camera using a Canon EF-S 10-18mm as lens and radially corrected with Adobe Lightroom (with settings for the mounted lens, using the strongest correction). We partitioned the image into non-overlapping annuli of width 64 pixels and found for each annulus—through exhaustive search—the value of α that maximizes $\text{PCE}_{\text{inv}}(\alpha)$ in (13), where inversion is done via (16). The result is plotted in Fig. 1 as a function of the inner radius of the annulus. As it is quite apparent, there is a dependence of α with r that indicates that one parameter alone is not sufficient to describe the radial transformation and that a more intricate relationship—even if parametric—must be sought.

III. State of the Art

The PCE is very sensitive to the correct alignment of the locations corresponding to the estimated PRNU and the residual; this means that unless a value of $\boldsymbol{\alpha}$ very close to the true one is used in the mappings in (12) or (13), the resulting PCE will be very small, and hypothesis H_1 is likely to be rejected when it is in force. To illustrate this phenomenon, in Fig. 2 we show the function $\text{PCE}_{\text{max}}(\alpha)$ for an image taken with a Panasonic DMC-ZS7 camera, shutter speed: 1/400 s, aperture: f4.4, focal length: 19.5 mm, and ISO 100. The stepsize in α is $2 \cdot 10^{-3}$. As we can observe, under H_1 the function is very spiky, with the consequence that a sufficiently dense grid must be used; otherwise, it is easy to miss the peak. In addition, this spikiness precludes the use of gradient-based algorithms, because they would only work in the very close vicinity of the peak.

Therefore, any search grid in the parameter space has to be fine enough to be able to locate the maximum. The method in [8] considers that the transformations (both the direct and the inverse) are parameterized by a scalar α and starts by selecting a search interval $[-A, A]$ which is progressively made finer so that at each iteration k, with $k = 1, \cdots, k_{\text{max}}$, a grid with $2^k + 1$ points is generated. Note that at the $k+1$-th iteration only 2^k new points are produced. A threshold τ_1 is set so that if, after all k_{max} iterations, no α exists in the grid such that $\text{PCE}_{\text{max}}(\alpha) > \tau_1$, then the search is stopped and a mismatch is declared (i.e., H_0 is decided). At every iteration, PCE_{max} is maximized over all grid points; this requires computing it only for the new points. Let α° denote the grid point for which the maximum is obtained; if at some iteration $\text{PCE}_{\text{max}}(\alpha^\circ) > \tau_1$, then the search stops and the algorithm proceeds to the second stage in order to refine the value of α°. However, in order to speed up the process, the maximization skips the exhaustive enumeration of all grid points provided that $k > 4$ whenever $\alpha^\dagger$ is found such that $\text{PCE}_{\text{max}}(\alpha^\dagger) > \tau_2$ (with $\tau_2 > \tau_1$). In this case, the algorithm proceeds to the second stage by searching around $\alpha^\dagger$. The second stage takes the value of α with which the first stage was exited and constructs an interval with its two neighboring points in the grid. If k^* is the exit value of k for the first stage, then this interval has width $A/2^{k^*-1}$. Next, a golden section search is performed until the width of the interval is approximately $1/(8D_2)$, with D_2 the half-diagonal of the image. Let α^* be the value found with the golden section search; then, if $\text{PCE}_{\text{max}}(\alpha^*) > \tau_3$ hypothesis H_1 is accepted, else, H_0 is declared. The thresholds suggested in [8] are $\tau_1 = 15$ and $\tau_2 = \tau_3 = 75$, and $k_{\text{max}} = 7$. To reduce the computational load [8] downsamples the signals by a factor of

two in each dimension; since this has an impact on accuracy in some cases, in the experimental section, we will consider both the downsampled (DS) and non-downsampled versions.

The method in [9] takes a different approach to perform the inversion of radially-corrected barrel distortions by employing the so-called linear patterns that are present in the residuals and are due to artifacts of the capturing device. These patterns are typically removed towards source attribution, but when kept, they serve as pilot signals that may be used to infer the radial distortion correction. The feature that is used to steer the parameter estimation is the energy of the linear pattern, defined for a given residual $\mathbf{W}$ as $E(\mathbf{W}) \doteq ||\mathbf{c}||^2 + ||\mathbf{r}||^2$, where $\mathbf{c}$ and $\mathbf{r}$ are vectors containing respectively the column and row averages of $\mathbf{W}$. Then, considering the set of fourth-order transformations $T_{\boldsymbol{\alpha}}(r) = r(1 + \alpha_2 r^2 + \alpha_4 r^4)$, where $\boldsymbol{\alpha} = (\alpha_2, \alpha_4)$, the method in [9] seeks to maximize $E(T_{\boldsymbol{\alpha}}^{-1}(\mathbf{W}))$ with respect to $\boldsymbol{\alpha}$, with the rationale that when the correct inverse transformation is applied, the linear pattern is recovered; otherwise, the column and row averages will be expected to produce low values. The fact that the transformation is now parameterized by two variables α_2 and α_4 gives more flexibility in inverting the transformation, but potentially incurs a larger computational cost. To make the optimization more manageable, a first stage consists in fitting a second-degree polynomial on variable α_2 to values of $E(T_{\boldsymbol{\alpha}}^{-1}(\mathbf{W}))$ sampled on a grid for $\alpha_2 \in [\alpha_{\min}, \alpha_{\max}], \alpha_{\min} > 0$, and $\alpha_4 = 0$. The reason for this choice of α_4 is that in practice the contribution of α_4 to $T_{\boldsymbol{\alpha}}(r)$ is only significant for large r, that is, far from the image center. This first stage yields the value $\alpha_2^{(1)}$ of α_2 that maximizes the difference from the energy of the linear pattern and its polynomial fit. The second stage employs a Nelder-Mead optimization (using the linear pattern energy as cost function) that is initialized with three points derived from $\alpha_2^{(1)}$. This produces the two optimal radial correction parameters (α_2^*, α_4^*). Due to noise, the previous procedure will yield an optimum $\alpha_2 \neq 0$ regardless of whether radial correction was applied. Then, the decision is confirmed only if the cost function evaluated in a neighborhood of (α_2^*, α_4^*) corroborates the existence of a significant peak; otherwise, the image is deemed to be not radially corrected.

Even though, as we will see in Sect. V, the performance of the two methods outlined above is rather good, they have two main intrinsic limitations that we aim at overcoming with our work: 1) their corresponding first stages employ an exhaustive search on a *fixed* grid. This fact, together with the high sensitivity of the PCE with respect to changes in the parameter vector $\boldsymbol{\alpha}$ about the correct one that results in a very spiky objective function, advise the use of a relatively tight grid to minimize the risk of missing the optimum. Unfortunately, this tightness entails a significant computational cost. 2) Again, due to the computational cost of an exhaustive search, the transformations $T_{\boldsymbol{\alpha}}$ and $T_{\boldsymbol{\alpha}}^{-1}$ use a small number of parameters: one in [8], and two in [9]. Therefore, these parameterization are unable to capture more complex radial corrections, such as those employed by editing programs (cf. Fig. 1), a trend that is likely to increase, as the capabilities of out-of-camera processing improve. A combination of the two methods in [8] and [9] has been recently proposed by the authors in [20].

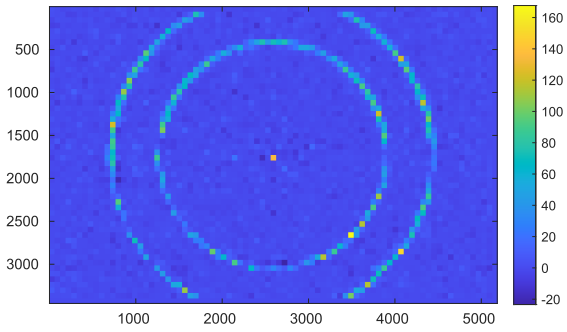

Fig. 3. $\text{PCE}_{\text{inv}}(\alpha)$ for: $\alpha = -0.01$ and $\alpha = 0.05$.

IV. Proposed Method

In order to motivate the method proposed in this paper, we will rely on an example generated with the popular photo editing software *Adobe Lightroom* that will give us the necessary clues. Images were taken with a Canon 1200D camera and then radially corrected with Lightroom. In Fig. 3 we combined two PCE_{inv} maps (corresponding to $\alpha = -0.01$ and $\alpha = 0.05$) in which $\text{PCE}_{\text{inv}}(\alpha)$ is computed using (13) and (16) for non-overlapping blocks of size 64×64. The combination is carried out by substituting the pixels corresponding to the outermost annulus (obtained for $\alpha = -0.01$) into the PCE_{inv} map obtained for $\alpha = 0.05$. For mere illustrative purposes, and in order to enhance the visibility, the (radially corrected) image under analysis (from which $\mathbf{W}$ is computed) is one of the 20 flat-field images used to extract $\hat{\mathbf{K}}'$. As we can see, the region where the PCE is significant is an annulus, and the position of the annulus depends on α. This shows that if $L(r)$ denotes the radial correction induced by the software and $L^{-1}(r)$ its inverse, then for a given $\alpha = \alpha_0$, $T_{\alpha_0}^{-1}(r) \approx L^{-1}(r)$ only in a small neighborhood of some $r = r^*$.

This experiment clearly indicates that for complex radial corrections, an approach like (16) will not work. However, the fact that the inversion works locally suggests breaking the problem into non-overlapping concentric annuli as shown in Fig. 4, and solving each separately.

A. Set Partitioning and Transform Computation

Let $\mathcal{R}_k$, $k = 1, \cdots, L$, be the kth annulus described by an inner radius r_k (recall that radii are scaled by D_2 so that $r = 1$ corresponds to half of the image diagonal) and a width Δ_k as follows:

$$\mathcal{R}_k \doteq \{(u, v) \in \mathbb{R}^2 : r_k^2 \leq u^2 + v^2 < (r_k + \Delta_k)^2\}. \quad (17)$$

The inner radii are generated as $r_{k+1} = r_k + \Delta_k$, with $r_1 = 0$, and the inner radius of the last annulus r_L is such that $r_L < 1 < r_L + \Delta_L$ (see Fig. 4). This definition implies that the first annulus degenerates into a disk and the image is fully covered by annuli. Except for this degenerate annulus, in this work we will assume that $\Delta_k = \Delta$ for all k.

The experiment shown in Fig. 1 (obtained applying a brute force search for each annulus) suggests that a good modeling of the radial correction can be obtained by allowing α to vary with r, so (10) in this case becomes

$$r' = T_{\alpha(r)}(r) = r(1 + \alpha(r) \cdot r^2). \quad (18)$$

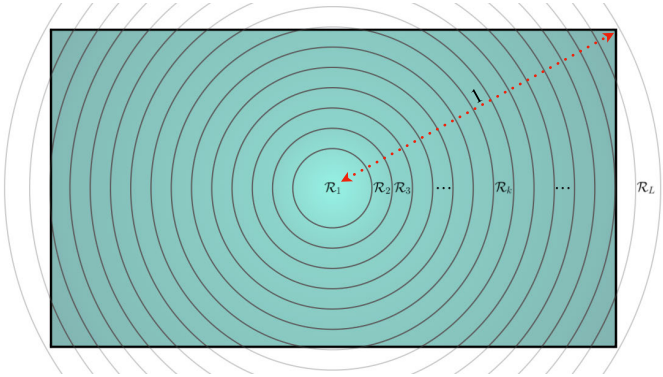

Fig. 4. Annular partition used in the proposed method.

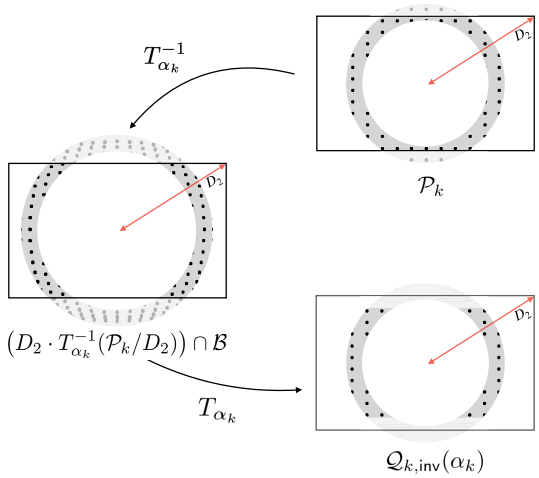

Fig. 5. Illustration of the application of transforms $T_{\alpha_k}^{-1}$ and T_{α_k}, and related domains.

The idea is that by allowing α to be a function of r, we achieve much more flexibility in modeling complex distortions. Moreover, as long as the annuli are thin enough, the zero-th order approximation $\alpha(r) \approx \alpha(r_k + \Delta_k/2) \doteq \alpha_k$ will be reasonably good for all $r \in \mathcal{R}_k$. This local approximation will allow us to use (16) for the inverse transform. However, since we are allowing α to vary with r, instead of a locally cubic dependence, as in (18), it also makes sense to consider a locally linear one, i.e., $r' = T_{\alpha(r)}(r) = r(1 + \alpha(r))$. Even though for the generic mappings we will keep using $T_{\alpha_k}(r)$ and $T_{\alpha_k}^{-1}(r)$ for the sake of generality, we specialize them by adding the sub-indices c to denote cubic, and l to denote linear. Therefore, on each annulus we write

$$\begin{aligned} T_{\alpha_k,c}(r) &\doteq r(1 + \alpha_k r^2); \\ T_{\alpha_k,l}(r) &\doteq r(1 + \alpha_k), \quad r \in \mathcal{R}_k, \end{aligned} \tag{19}$$

whereas the corresponding inverse mappings are

$$\begin{aligned} T_{\alpha_k,c}^{-1}(r') &\approx r'(1 - \alpha_k r'^2 + 3\alpha_k^2 r'^4), \quad r' \in T_{\alpha_k,c}(\mathcal{R}_k); \\ T_{\alpha_k,l}^{-1}(r') &= \frac{r'}{1 + \alpha_k}, \quad r' \in T_{\alpha_k,l}(\mathcal{R}_k), \end{aligned} \tag{20}$$

Note that the ranges of the inverse transforms in (20) may be different because the image of each annulus will differ under the locally cubic and locally linear mappings.

Given a collection of annuli $\mathcal{R}_k$, $k = 1, \cdots, L$, one can see the mapping $T_{\boldsymbol{\alpha}}(r)$ in (12) as a sequence of transformations $T_{\alpha_k}(r)$, $k = 1, \cdots, L$, that is parameterized by a vector $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_L]^T$. This larger number of parameters provides a much higher capacity of expressing complex radial functions, as required by our targeted distortion compensations. Obviously, the maximization of the PCE with respect to $\boldsymbol{\alpha} \in \mathcal{A} \doteq \mathcal{A}_1 \times \cdots \mathcal{A}_L$, with $\mathcal{A}_k$ the feasible set for α_k, would suffer from a combinatorial explosion due to the L dimensions involved, so we will be interested in finding efficient alternative ways for performing an approximate maximization.

A first step is to treat each annulus separately and find the optimal value of α_k constrained to the kth annulus. There are several possible approaches at this stage. One would be to find α_k that maximizes the PCE constrained to the kth annulus; unfortunately, since the total PCE *is not* the sum of those constrained PCEs, it is quite difficult to work individually with each annulus using such a criterion. Instead, we have opted for a maximum likelihood estimation approach that aims at finding the α_k that has the highest likelihood of producing the observed cross-correlations with the estimated PRNU. Once we describe how the optimal α_k can be found for each annulus in an adaptive way (Sect. IV-C), we proceed by explaining how the PCE can be computed and updated (Sect. IV-D).

In the following, we give a formal description of the annuli for the inverse approach (i.e., using $T_{\alpha_k}^{-1}$) and afterwards indicate how to adapt the discussion to the direct approach. Let $\mathcal{P}_k$ be the set of points of the image grid that are contained in the kth annulus, i.e.,

$$\mathcal{P}_k \doteq (D_2 \cdot \mathcal{R}_k) \cap \mathcal{I}, \quad k = 1, \cdots, L, \tag{21}$$

where multiplication of $\mathcal{R}_k$ by D_2 (i.e., half the diagonal in pixels) is necessary to re-scale the annulus back to integer-valued coordinates (recall that $r = 1$ corresponds to half the diagonal).

Given $\tilde{\mathbf{W}} = \mathbf{W} - \bar{\mathbf{W}}$ and $\mathcal{P}_k$, computation of $T_{\alpha_k}^{-1}(\tilde{\mathbf{W}})$ proceeds as follows (see Fig. 5). First, the image of the set $\mathcal{P}_k$ under $T_{\alpha_k}^{-1}$, i.e. $T_{\alpha_k}^{-1}(\mathcal{P}_k)$, is calculated and the transformed points lying outside the image boundaries $\mathcal{B}$ are discarded, as the subsequent interpolation would not be computable. For the remaining points, $T_{\alpha_k}^{-1}(\tilde{\mathbf{W}})$ is obtained by interpolation from $\tilde{\mathbf{W}}$. We let $\mathcal{Q}_{k,\mathsf{inv}}(\alpha_k)$ be the set of points of $\mathcal{P}_k$ for which their image under $T_{\alpha_k}^{-1}$ exists (the sub-index inv stands for 'inverse approach'). Formally, this set is

$$\mathcal{Q}_{k,\mathsf{inv}}(\alpha_k) = D_2 \cdot T_{\alpha_k}\left(\left[\left(D_2 \cdot T_{\alpha_k}^{-1}(\mathcal{P}_k/D_2)\right) \cap \mathcal{B}\right]/D_2\right). \tag{22}$$

Notice that if the set $\mathcal{P}_k$ transformed via $T_{\alpha_k}^{-1}$ does not get out of the image bounds $\mathcal{B}$, then $\mathcal{Q}_{k,\mathsf{inv}}(\alpha_k) = \mathcal{P}_k$; otherwise, $\mathcal{Q}_{k,\mathsf{inv}}(\alpha_k) \subset \mathcal{P}_k$. As a consequence, $\mathcal{Q}_{k,\mathsf{inv}}(\alpha_k) \cap \mathcal{P}_k = \mathcal{Q}_{k,\mathsf{inv}}(\alpha_k)$. Also notice that, as explicitly indicated, the set $\mathcal{Q}_{k,\mathsf{inv}}(\alpha_k)$ and, in particular, its cardinality, varies with α_k.

For the direct approach, the considerations are similar. Basically, we have to exchange the roles of T_{α_k} and $T_{\alpha_k}^{-1}$. Recalling that the sub-index dir stands for 'direct approach',

the set $\mathcal{Q}_{k,\mathsf{dir}}(\alpha_k)$ can be formally written as

$$\mathcal{Q}_{k,\mathsf{dir}}(\alpha_k) = D_2 \cdot T_{\alpha_k}^{-1}\left(\left[\left(D_2 \cdot T_{\alpha_k}(\mathcal{P}_k/D_2)\right) \cap \mathcal{B}\right]/D_2\right). \tag{23}$$

B. Optimization With Respect to α_k

Once the annuli have been characterized, in this section we address the problem of finding the optimal values of α_k that parameterize the transformations $T_{\alpha_k}^{-1}$ and T_{α_k} for the kth annulus.

For the sake of compactness, we will find it useful to denote the cross-correlation and the energy of the transformed residual computed over $\mathcal{Q}_{k,\mathsf{inv}}(\alpha_k)$ as, respectively,

$$\Phi_{k,\mathsf{inv}}(\alpha_k) \doteq \sum_{(i,j)\in\mathcal{Q}_{k,\mathsf{inv}}(\alpha_k)} \hat{K}'_{i,j} \cdot \left[T_{\alpha_k}^{-1}(\tilde{\mathbf{W}})\right]_{i,j}, \tag{24}$$

$$\mathsf{E}_{k,\mathsf{inv}}(\alpha_k) \doteq \sum_{(i,j)\in\mathcal{Q}_{k,\mathsf{inv}}(\alpha_k)} \left[T_{\alpha_k}^{-1}(\tilde{\mathbf{W}})\right]_{i,j}^2, \tag{25}$$

by making implicit the use of the inverse transformation $T_{\alpha_k}^{-1}(\cdot)$, and $\hat{\mathbf{K}}'$ and $\tilde{\mathbf{W}}$. Similarly, we denote by $\Phi_{k,\mathsf{dir}}(\alpha_k)$ and $\mathsf{E}_{k,\mathsf{dir}}(\alpha_k)$ the cross-correlation and energy for the direct mapping $T_{\alpha_k}(\cdot)$ computed over $\mathcal{Q}_{k,\mathsf{dir}}(\alpha_k)$.

In Appendix A we derive an estimator of α_k on the kth annulus. This estimator is rooted in the principle of maximum likelihood applied to the output of a bank of cross-correlations. For the inverse approach, this becomes

$$\alpha_k^* = \arg\max_{\alpha_k\in\mathcal{A}_k} \varphi_{k,\mathsf{inv}}(\alpha_k), \tag{26}$$

where

$$\varphi_{k,\mathsf{inv}}(\alpha_k) \doteq \frac{\Phi_{k,\mathsf{inv}}(\alpha_k)}{\mathsf{E}_{k,\mathsf{inv}}(\alpha_k)}. \tag{27}$$

For the direct approach, the optimization is carried out after replacing the subindex inv by dir in both (26) and (27). We notice the proposed objective function is different from the PCE (constrained to the kth annulus); besides the theoretical justification in Appendix A, in [21] we provide empirical evidence that optimization of our objective function renders better global performance than the PCE.

C. Adaptive Optimization

One key observation from Fig. 1 is that the sequence α_k^*, $k = 1, \cdots, L$, changes smoothly for sufficiently small Δ_k. This hints at the possibility of reducing the computational complexity of the exhaustive search by using an adaptive predictor. In our case, we will show experimentally that a linear predictor $\mathbf{u}$ with length U suffices to achieve excellent results. In the following, we explain this adaptive procedure. As above, we will give the details for the inverse approach, as the direct one is methodologically identical.

First, we need to select an initial index that we will denote by k_0. To this end, we look for the annulus that gives the best results under no transformations (i.e., when $\alpha_k = 0$). Formally, this implies that

$$k_0 = \arg\max_{k=1,\cdots,L} \varphi_{k,\mathsf{inv}}(0). \tag{28}$$

Once this initial point is found, the optimal value of α_{k_0} is found by exhaustive search in a discrete set around $\alpha_{k_0} = 0$. Let $\mathcal{A}_{k_0}$ be such a neighborhood, then following (26), $\alpha_{k_0}^* = \arg\max_{\alpha_{k_0}\in\mathcal{A}_{k_0}} \varphi_{k_0,\mathsf{inv}}(\alpha_{k_0})$.

We will find it useful to define an auxiliary sequence $\{\beta_k\}$ that is initialized as $\beta_k = \alpha_{k_0}^* \cdot \delta_{k-k_0}$, where δ_k is Kronecker's delta.[1] This sequence is used to store the regressor values. Since the starting point is $k = k_0$, there are two possible directions for the prediction: forward (i.e., $k > k_0$), and backward (i.e, $k < k_0$).[2] We will describe how the former is carried out, and then indicate the modifications needed for the latter. We define the forward regressor at index k as $\boldsymbol{\beta}_k^T \doteq [\beta_{k-U+1}, \cdots, \beta_{k-1}, \beta_k]$, where U is the length. Notice that, from the way the auxiliary sequence is initialized, $\boldsymbol{\beta}_{k_0}^T = [0, \cdots, 0, \alpha_{k_0}^*]$. We also need a vector of weights at index k that will be denoted by $\mathbf{u}_k$; this vector of length U is initialized as $\mathbf{u}_{k_0}^T = [0, \cdots, 0, 1]$. Then, for $k > k_0$ the output of the predictor at index k will be computed as

$$\hat{\alpha}_k = \mathbf{u}_{k-1}^T \boldsymbol{\beta}_{k-1}, \tag{29}$$

for $k = k_0 + 1, \cdots, L$. This predicted value is refined by exhaustive search in a discrete neighborhood of $\hat{\alpha}_k$. Let $\mathcal{A}_k$ denote such a neighborhood; then α_k^* is obtained as in (26). The details on how the neighborhood $\mathcal{A}_k$ is constructed are given below. Before that, we explain the updating procedure for $\mathbf{u}_k$ and $\boldsymbol{\beta}_k$. To that end, we define the *a posteriori error* at index k as

$$e_k \doteq \alpha_k^* - \hat{\alpha}_k, \tag{30}$$

for $k = k_0 + 1, \cdots, L$. This error is used to drive the adaptive algorithm. It is easy to show that the gradient vector of $|e_k|^2$ with respect to the weights vector $\mathbf{u}_{k-1}$ is equal to $-2e_k\boldsymbol{\beta}_{k-1}$. Then, following the Least Mean Squares algorithm [22], we propose to update the weights by taking a step in the direction of the negative gradient, that is,

$$\mathbf{u}_k = \mathbf{u}_{k-1} + \mu e_k \boldsymbol{\beta}_{k-1}, \quad k = k_0 + 1, \cdots, L, \tag{31}$$

where μ is the so-called step-size. The update of the sequence $\{\beta_k\}$ containing the regressor is done by making $\beta_k = \alpha_k^*$; the forward regressor vector $\boldsymbol{\beta}_k$ is updated accordingly. This iterative procedure is then repeated by going back to (29) and proceeding until the sequence $\alpha_{k_0+1}^*, \alpha_{k_0+2}^*, \cdots, \alpha_L^*$ is produced.

The backward prediction proceeds in a similar way, but now vector $\boldsymbol{\beta}_k$ is defined as $\boldsymbol{\beta}_k^T \doteq [\beta_k, \beta_{k+1}, \cdots, \beta_{k+U-1}]$; this means that at the backward initialization, vector $\boldsymbol{\beta}_{k_0}$ will take advantage of the availability of values of α_k^* that have been already computed, i.e., $\boldsymbol{\beta}_{k_0} = [\alpha_{k_0}^*, \alpha_{k_0+1}^*, \cdots, \alpha_{k_0+U-1}^*]^T$. The weights vector for the backward prediction $\mathbf{u}_{k_0}$ is initialized as $\mathbf{u}_{k_0} = [1, 0, \cdots, 0]$. Now this weights vector is updated in the reverse direction:

$$\mathbf{u}_k = \mathbf{u}_{k+1} + \mu e_k \boldsymbol{\beta}_{k+1}, \quad k = k_0 - 1, \cdots, 1, \tag{32}$$

[1] Although from a notational point of view, it would be more correct to define a sequence for every iteration of the algorithm, we allow replacing values in this sequence in order to avoid overcomplicating the notation.

[2] Degenerate cases arise when $k_0 = L$ or $k_0 = 1$, for which the forward and backward predictions, respectively, are not needed.

and again the sequence $\{\beta_k\}$ containing the regressor is updated by making $\beta_k = \alpha_k^*$; the backward regressor vector $\boldsymbol{\beta}_k$ is updated accordingly. The algorithm thus generates the sequence $\alpha^*_{k_0-1}, \alpha^*_{k_0-2}, \cdots, \alpha^*_1$.

After both forward and backward predictions are finished, the optimal vector is $\boldsymbol{\alpha}^*_{\mathsf{inv}} = [\alpha_1^*, \cdots, \alpha_L^*]^T \in \mathcal{A}$, where once again we have added the subindex inv to stress the fact that we are dealing with the inverse approach. The same procedure applied to the direct approach will yield an optimal vector $\boldsymbol{\alpha}^*_{\mathsf{dir}}$. The pseudo-code for the proposed algorithm is provided in the technical report [21].[3]

One critical point of the algorithm is the refining of $\hat{\alpha}_k$ that produces α_k^*. While smarter strategies might be possible, here we perform an exhaustive search around $\hat{\alpha}_k$ in a discrete set $\mathcal{A}_k$. Of course, the cardinality of this set must be kept at a small value in order to limit the computational burden. On the other hand, the discrete points must be generated finely enough to output a value that is sufficiently close to the optimal. We thus employ two parameters to describe the set: λ_k that controls the resolution, and A_k that is an odd integer that determines the number of points. Then, given $\hat{\alpha}_k$ and these parameters, the search set is constructed as:

$$\mathcal{A}_k = \{\hat{\alpha}_k + \lambda_k \cdot n : n \in \mathbb{Z} \cap [-(A_k-1)/2, (A_k-1)/2]\}. \tag{33}$$

Note that this construction guarantees that $|\mathcal{A}_k| = A_k$. The parameter λ_k is selected to be commensurate with $|\alpha_k^* - \alpha_{k-1}^*|$ in the forward case (resp. $|\alpha_k^* - \alpha_{k+1}^*|$ in the backward case), so that the smaller the change in α_k^*, the finer the grid. In Sect. IV-G we give more details about the rules that were employed to generate λ_k for the experiments. Regarding the size of the set A_k, this is updated in the same loop as the predictor; for the forward predictor, the rule is as follows: if for index k the maximum α_k^* is found at one of the extremes of the set $\mathcal{A}_k$ (i.e., $\alpha_k^* = \hat{\alpha}_k - \lambda_k \cdot (A_k-1)/2$ or $\alpha_k^* = \hat{\alpha}_k + \lambda_k \cdot (A_k-1)/2$) then the size of the set is increased at the following iteration, i.e., $A_{k+1} = A_k + 2$. Otherwise, if A_k is already small, i.e., $A_k = A_{\min}$ for some minimum size $A_{\min}$, then $A_{k+1} = A_{\min}$; else (i.e, if the maximum in $\mathcal{A}_k$ is not found at either of the extremes, and the set is large enough), the size is decreased at the following iteration, i.e., $A_{k+1} = A_k - 2$. This update is intended to find a compromise between the size of the set and the objective of capturing the optimal α_k. For the backward prediction the reasoning is identical, but updating A_{k-1} from A_k. An example of the evolution of α_k^* and $\mathcal{A}_k$ with respect to the different annuli is shown in Fig. 6 for a test image taken with a Canon 1200D camera, Canon EF-S 10-18mm lens, and corrected with Adobe Lightroom using the strongest radial correction. The reference PRNU is estimated with 20 natural images all obtained with the same settings but with no Lightroom correction. The star indicates the value of $\alpha^*_{k_0}$ estimated for the initial iteration, the red line indicates the forward prediction direction and the blue line the backward one. Finally, the band in pale blue encompasses the set $\mathcal{A}_k$ containing the candidate values of α_k for the kth annulus. The "Dir, Cub" variant of our algorithm is employed (see Sect. V).

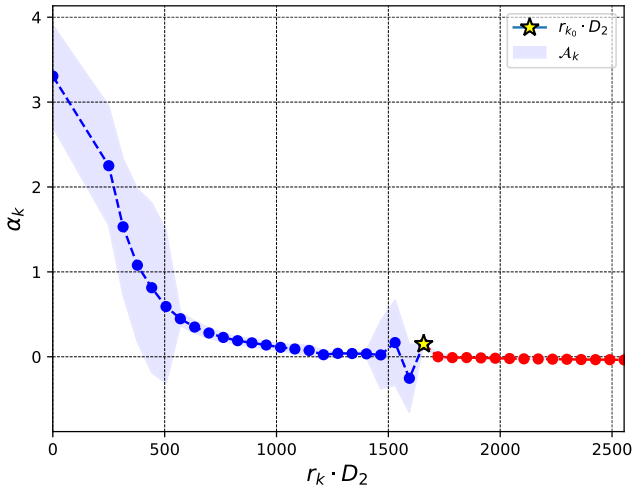

Fig. 6. Evolution of α_k^* and $\mathcal{A}_k$ versus $r_k \cdot D_2$. Test image taken with Canon 1200D camera, Canon EF-S 10-18mm lens, corrected with Adobe Lightroom. Focal length: 10mm. Shutter speed: 1/120 sec. Aperture: f8.0. ISO "Dir, Cub" variant.

D. PCE Computation for the Optimal $\boldsymbol{\alpha}$

As a result of the adaptive algorithm presented in the previous section, it is possible to compute the PCEs that are required in the hypothesis test, that is, $\mathrm{PCE}_{\mathsf{inv}}(\boldsymbol{\alpha}^*_{\mathsf{inv}})$ and $\mathrm{PCE}_{\mathsf{dir}}(\boldsymbol{\alpha}^*_{\mathsf{dir}})$, see the definitions in (12) and (13). In both cases, the numerator and denominator of the PCE are already available, as they are required for the optimization. The only additional computations are simple sums to accumulate the results corresponding to the different annuli. To see how this is so for the inverse approach, notice first that the right hand side of (13) requires computing the difference $T^{-1}_{\boldsymbol{\alpha}^*}(\mathbf{W}) - T^{-1}_{\boldsymbol{\alpha}^*}(\mathbf{W})$ (cf. the expression of the PCE in (6)), which can be simplified by noticing that: 1) It is reasonable to write $T^{-1}_{\boldsymbol{\alpha}^*}(\mathbf{W}) \approx T^{-1}_{\boldsymbol{\alpha}^*}(\bar{\mathbf{W}})$ because $T^{-1}_{\boldsymbol{\alpha}^*}$ is a geometrical transformation that will not substantially alter the mean value of the residual.[4] 2) Due to zero-meaning on the residual, it is possible to write $\bar{\mathbf{W}} = \mathbf{0}$. With these considerations, we can write $T^{-1}_{\boldsymbol{\alpha}^*}(\mathbf{W}) - T^{-1}_{\boldsymbol{\alpha}^*}(\mathbf{W}) \approx T^{-1}_{\boldsymbol{\alpha}^*}(\tilde{\mathbf{W}})$, which is simpler to compute.

With this approximation, the numerator of $\mathrm{PCE}(\hat{\mathbf{K}}', T^{-1}_{\boldsymbol{\alpha}^*}(\mathbf{W}))$ can be expanded as follows

$$\mathrm{ssq}(\langle \hat{\mathbf{K}}', T^{-1}_{\boldsymbol{\alpha}^*}(\tilde{\mathbf{W}})\rangle) = \sum_{k=1}^{L} \mathrm{ssq}(\Phi_{k,\mathsf{inv}}(\alpha_k^*)), \tag{34}$$

Now we can easily identify each of the L summands in (34) as the numerator of $\varphi_{k,\mathsf{inv}}(\alpha_k^*)$ in (27) which can be stored during the adaptive optimization process.

The denominator of the PCE requires more attention. With the approximation above, this denominator is $\frac{1}{|\mathcal{I}_T \setminus \mathcal{S}|} \sum_{\mathbf{s} \in \mathcal{I}_T \setminus \mathcal{S}} \langle \hat{\mathbf{K}}', C(T^{-1}_{\boldsymbol{\alpha}^*}(\tilde{\mathbf{W}}), \mathbf{s})\rangle^2$ which is nothing but a sample estimate of the variance of the cross-correlation of

[3]The code is available at https://github.com/AMontiB/AdaptivePRNUCameraAttribution

[4]Strict equality does not hold because $\bar{\mathbf{W}}$ and $T^{-1}_{\boldsymbol{\alpha}^*}(\bar{\mathbf{W}})$ do not have the same support.

$\hat{\mathbf{K}}'$ and $T_{\boldsymbol{\alpha}^*}^{-1}(\tilde{\mathbf{W}})$. In [21, Sect. VIII] we derive and discuss a simpler sample estimate that is more statistically efficient (i.e., has a lower variance). This fully justifies the approximation

$$\frac{1}{|\mathcal{I}_T \backslash \mathcal{S}|} \sum_{\mathbf{s} \in \mathcal{I}_T \backslash \mathcal{S}} \langle \hat{\mathbf{K}}', C(T_{\boldsymbol{\alpha}^*}^{-1}(\tilde{\mathbf{W}}), \mathbf{s}) \rangle^2 \approx \kappa \cdot \hat{\sigma}^2_{\hat{K}'} \cdot \sum_{k=1}^{L} \mathsf{E}_{k,\mathsf{inv}}(\alpha_k^*), \tag{35}$$

where $\hat{\sigma}^2_{\hat{K}'} \doteq ||\hat{\mathbf{K}}'||^2/|\mathcal{I}|$ (recall that $\hat{K}'_{i,j}$ exists for all $(i, j) \in \mathcal{I}$), and κ is a factor that takes into account the fact that the cardinalities of $\mathcal{I}$ and $\bigcup_{k=1}^{L} \mathcal{Q}_{k,\mathsf{inv}}(\alpha_k^*)$ are different. (In practice, κ will be close to 1, so it can be dropped.) Once again, the L summands in (35) are already available as the denominator of (27).

E. Early Stopping

The partition into annuli offers one remarkable byproduct: taking inspiration from [23], it is possible to stop processing annuli (and declare that H_1 holds) if a cumulative PCE exceeds a predefined threshold. Following the approximations in the previous subsection, one might be tempted to compute a cumulative PCE by using the numerators and denominators already produced during the optimization. In this way, the optimization would not need to be carried out for all annuli but instead it could be stopped as soon as the PCE computed so far exceeds the threshold. Unfortunately, this approach would be incorrect, because while a fraction with sums in the numerator can be expanded into a sum of fractions, this is not the case when there are sums in the denominator. Therefore, if we want to implement an early stopping mechanism, we need to seek ways to further approximate the denominator of the PCE without actually computing all the elements of $\boldsymbol{\alpha}^*$. To this end, we can ask ourselves how sensitive is the right hand side of (35) to changes in α_k^*; after all, since each of the L summands is an estimate of the variance of the transformed residual inside an annulus, one would expect not much variation for realistic values of α. If this were the case, then one might approximate the right hand side of (35) (which corresponds to the optimal vector $\boldsymbol{\alpha}_{\mathsf{inv}}^*$) by computing it for any reasonable value of $\boldsymbol{\alpha}_{\mathsf{inv}}$ without involving any optimization.

In order to illustrate the feasibility of this approximation, we show in Fig. 7 the values of the sample variance of a transformed residual computed in each annulus, i.e., $\hat{\sigma}^2_W \doteq \frac{\mathsf{E}_{k,\mathsf{inv}}(\alpha_k)}{|\mathcal{Q}_{k,\mathsf{inv}}(\alpha_k)|}$ as a function of α_k for several annuli (i.e., $k = 18, 22, 33$) and for cubic inverse mappings, see (20). Fig. 7 also shows the value of the variance estimated from the full-size transformed residual, i.e., $\frac{1}{|\mathcal{I}_T|}||T_{\alpha_k}^{-1}(\tilde{\mathbf{W}})||^2$. Bi-linear interpolation is used in all cases.

As we can see, the variance estimate is fairly constant for different values of α_k, except in a neighborhood of zero. Moreover, this is similar to the variance estimate obtained from the whole transformed residual, so the latter can be used in place of the variance estimate for a specific annulus. The reason for the spike at $\alpha_k = 0$ is that the interpolation that is needed for computing the inverse mapping when $\alpha_k \neq 0$ produces a reduction in the variance of the transformed residual. This reduction depends on the square magnitude of the interpolation

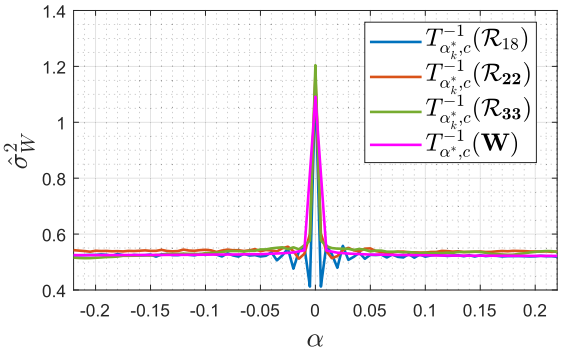

Fig. 7. Sample variance of the transformed residual for different annuli and different values of α. Camera and parameters are the same as in Fig. 1.

filter at different sampling points. In general, the grids before and after the interpolation are not related through rational numbers, but for certain annuli and values of α, moiré patterns between the sampling grids may appear; this is why in Fig. 7 a ripple near zero is observed for the annuli $k = 18, 22$. The energy reduction phenomenon has been reported in [24] in a different scenario but related to ours.

The invariance discussed in the previous paragraph suggests several ways of approximating the right hand side of (35); for instance, it is possible to pick any value of α, say α_{f}, sufficiently far from $\alpha = 0$ and for all the annuli use the same transformation $T_{\alpha_{\mathsf{f}}}^{-1}$ in place of $T_{\alpha_k^*}^{-1}$. We remark that the reason why the neighborhood of $\alpha = 0$ should be excluded when selecting α_{f} is the fact that inside such a neighborhood the denominator of the PCE is overestimated and, consequently, the PCE underestimated.

Another way of approximating the right hand side of (35) which offers a slightly better performance than the former is to use the values of α_k^* already available from the optimization to update the approximation. This comes at practically no cost because the corresponding term $\mathsf{E}_{k,\mathsf{inv}}(\alpha_k^*)$ needs to be computed anyway during the optimization. For those annuli whose α_k^* is not available yet, the corresponding term is substituted by its approximation computed at $\alpha_k = \alpha_{\mathsf{f}}$.

We explain next how to compute the Cumulative PCE at the nth iteration which we will denote by $\mathrm{CPCE}_{n,\mathsf{inv}}(\hat{\mathbf{K}}', \mathbf{W})$. First, we need a mapping $\xi : \{1, \cdots, L\} \to \{1, \cdots, L\}$, from the natural order to the one induced by the proposed iterative procedure, i.e., $\xi(1) \mapsto k_0, \xi(2) \mapsto k_0 + 1, \cdots, \xi(L - k_0) \mapsto L, \xi(L - k_0 + 1) \mapsto k_0 - 1, \cdots, \xi(L) \mapsto 1$. Then,

$$\mathrm{CPCE}_{n,\mathsf{inv}}(\hat{\mathbf{K}}', \mathbf{W}) \doteq \frac{\sum_{k=\xi(1)}^{\xi(n)} \mathrm{ssq}\left(\Phi_{k,\mathsf{inv}}(\alpha_k^*)\right)}{\hat{\sigma}^2_{\hat{K}'} \left(\sum_{k=\xi(1)}^{\xi(n)} \mathsf{E}_{k,\mathsf{inv}}(\alpha_k^*) + \sum_{k=\xi(n+1)}^{\xi(L)} \mathsf{E}_{k,\mathsf{inv}}(\alpha_{\mathsf{f}}) \right)}. \tag{36}$$

Thus, the early-stopping algorithm will declare a match and stop if for some $n = 1, \cdots, L$, $\mathrm{CPCE}_{n,\mathsf{inv}}(\hat{\mathbf{K}}', \mathbf{W}) > \tau_c$ is satisfied. The value of τ_c is set experimentally to achieve the desired False Positive Rate (FPR).

Given the numerator and denominator of $\mathrm{CPCE}_{n,\mathsf{inv}}(\hat{\mathbf{K}}', \mathbf{W})$, and once $\alpha_{\xi(n+1)}^*$ is available, the numerator of $\mathrm{CPCE}_{n+1,\mathsf{inv}}(\hat{\mathbf{K}}', \mathbf{W})$ is updated by adding

$\Phi_{\xi(n+1),\mathsf{inv}}(\alpha^*_{\xi(n+1)})$, while the update of the denominator requires adding $\mathsf{E}_{\xi(n+1),\mathsf{inv}}(\alpha^*_{\xi(n+1)})$ and subtracting $\mathsf{E}_{\xi(n+1),\mathsf{inv}}(\alpha_{\mathsf{f}})$.

A similar definition follows for the Cumulative PCE in the direct approach $\mathrm{CPCE}_{n,\mathsf{dir}}(\hat{\mathbf{K}}', \mathbf{W})$ and the corresponding early stopping criterion.

F. Parameter Inheritance

As we have discussed, the test decision statistic takes the maximum of the PCEs computed through the direct and the inverse approaches. This implies that it is necessary to compute the optimal vector $\boldsymbol{\alpha}^*$ for both approaches, so the computational complexity is roughly doubled. This also holds if the early stopping criterion introduced above is applied. In such a case, the iterations for both the direct and the inverse approaches are made in parallel, so that for every k both $\mathrm{CPCE}_{n,\mathsf{dir}}(\hat{\mathbf{K}}', \mathbf{W})$ and $\mathrm{CPCE}_{n,\mathsf{inv}}(\hat{\mathbf{K}}', \mathbf{W})$ are checked against the threshold in order to stop as early as possible.

There is one sub-optimal way to alleviate the computational burden due to keeping the two approaches. We term it *parameter inheritance* and basically consists in using for the direct approach the same vector $\boldsymbol{\alpha}^*$ that was computed for the inverse approach. Of course, the latter is not necessarily optimal for the direct approach, but the rationale is that inside each annulus $\mathcal{R}_k$ the direct and inverse transformations nearly correspond to each other for the same value of α_k. Perfect correspondence does not exist because the inverse transformation is only an approximation and due to the fact that the search algorithm is prone to errors due to noise and insufficient resolution.

G. Parameter Default Values

In this section we provide the default values for the parameters of our algorithm and discuss some decisions regarding the initialization. These default values were used in the experiments reported in Sect. V. Specifically, the radius of the inner disk r_1 is such that $r_1 \cdot D_2$ equals 250 pixels and the width of each annulus Δ_k is such that $\Delta_k \cdot D_2$ equals 64 pixels. Both values are chosen as a compromise between performance and computational cost; see Sect.V for an additional discussion. For the linear predictor we set $U = 6$, $\mu = 1$ and $A_{\mathsf{min}} = 7$.

The initial search set $\mathcal{A}_{k_0}$ is given by $\mathcal{A}_{k_0} = \{-0.22, -0.21, \cdots, 0.21, 0.22\}$, which is the same range as used and justified in [8] to cover a variety of barrel and pincushion distortions. However, in our case we apply a coarser resolution for computational reasons and because the adaptive nature of our algorithm automatically adjusts to finer resolutions after a few iterations. We are aware that in [9] a wider range was preferred (even if just to invert pincushion distortions), so we carried out some experiments with images taken with the Canon 1200D camera and radially corrected with Adobe Lightroom using the lens distortion model of a different device (see Section V), since this combination produces some of the strongest and most variable radial corrections of our dataset. In these experiments, the search set was expanded to $\mathcal{A}_{k_0} = \{-0.50, -0, 49, \cdots, 0.49, 0.50\}$. While it is true that this set allows in some cases to get closer to the proper α_{k_0}, we found no significant differences in terms of performance with the previous initialization; as mentioned, this is due to our algorithm quickly finding the right range for α_k after few iterations. In contrast, the computational load of using the enlarged search set would be larger; for this reason, we recommend $\mathcal{A}_{k_0} = \{-0.22, -0.21, \cdots, 0.21, 0.22\}$. For an in-depth complementary discussion on the initial set, please see [21].

After the initial search, for the forward prediction $\mathcal{A}_{k_0+1}$ is given by (33) with $\lambda_{k_0+1} = 0.001$ and $A_{k_0+1} = 9$. For the following iterations,

$$\lambda_k = \begin{cases} 0.1 & \text{if } |\alpha_k - \alpha_{k-1}| > 0.1, \\ 0.01 & \text{if } 0.01 < |\alpha_k - \alpha_{k-1}| \leq 0.1, \\ 0.001 & \text{if } |\alpha_k - \alpha_{k-1}| \leq 0.001. \end{cases} \tag{37}$$

Identical considerations to the previous paragraph are made in regard to the backward prediction, where $k_0 + 1$ is replaced now by $k_0 - 1$ and in (37) $k - 1$ is replaced by $k + 1$.

V. Experimental Results

A. Experiments With in-Camera and Out-Camera Corrections

In order to measure the performance of the methods presented in Sect. IV and compare them with the state of the art in [8] and [9], we built a test dataset composed of 3645 images, of which 2037 were taken with the following compact cameras and radially corrected "in-camera" (i.e., by the camera software): Canon SX230 HS (188 images), Panasonic ZS7 (170 images), Canon SX40 (57 images), Canon SX210 (82 images), and Nikon S9100 (1540 images). All these images were downloaded from Flickr, as done in [8] and [9]; for this reason, there is an uneven distribution of images per device. 1508 of the remaining images in the test dataset were taken with the Canon 1200D (a reflex camera not applying any type of in-camera post-processing) with the following Canon Zoom Lenses: 1) EF-S 10-18 mm 1:4-5.6 IS STM; 2) EF-S 18-55 mm 1:3.5-5.6; 3) EF 75-300 mm 1:4-5.6, all radially corrected "out-camera" with third-party editing software: Adobe Lightroom Classic CC 2017, Adobe Photoshop CC 2017, PT Lens v2.0 (Macbook) and Gimp 2.10.14. Specifically, 377 images were corrected with each of these tools. With Adobe Lightroom we applied the correction model specific to the lens used to take the picture, thanks to the database of radial correction models Lightroom is equipped with. For the other editing software we applied the strongest radial correction available, as those tools cannot be tuned to a specific lens model. The last 100 images in the test dataset were also taken with the Canon 1200D camera but corrected with Lightroom using models for other lenses (i.e., Nikon, Tamron, Apple, Huawei and DJI, with 20 images each), always applying the strongest radial correction. This latter subset will be labeled as "Lightroom*" in the following.

Images in the test dataset were JPEG compressed with QFs in the range 90-98. For each device, the same QF is consistently used; see [21] for details. The reference PRNUs for carrying out the tests were estimated for each device using (4) with $L = 20$ natural images (not used for testing) compressed

with matching QFs to the test subset of that device. For the compact devices, since the in-camera corrections depend on the focal length, fixed specific values of the latter were sought in order to estimate the respective PRNUs; whenever enough images were available for a certain device and focal length, a different fingerprint was estimated and the results averaged for each device. In all cases, hypothesis H_1 was tested with images taken with focal lengths different from those used to estimate the fingerprints. We refer the reader to [21] for full details. When, under hypothesis H_0, the test images and the fingerprints have different sizes, we crop the central part of the larger to match its size to the smaller [7].

Next, we describe the identifiers used to refer to the different variants of our method in the figures and tables in this section. With "Dir" and "Inv" we indicate those cases where $\text{CPCE}_{n,\text{dir}}(\hat{\mathbf{K}}', \mathbf{W})$ and $\text{CPCE}_{n,\text{inv}}(\hat{\mathbf{K}}', \mathbf{W})$ are respectively used as the only test statistics. By "2W" we refer to the "two-way" case in which both the direct and the inverse approaches are used and H_1 is decided if either $\text{CPCE}_{n,\text{dir}}(\hat{\mathbf{K}}', \mathbf{W})$ or $\text{CPCE}_{n,\text{inv}}(\hat{\mathbf{K}}', \mathbf{W})$ are above the threshold for any $n \in \{1, \cdots, L\}$. To alleviate the computational load of the "two-way" parameter optimization, recall that in Sect.IV-F we proposed to inherit the parameters of one approach to the other. We will use the label $\overrightarrow{\text{DI}}$ to indicate inheritance of α_n^* from the direct approach to the inverse one; and $\overrightarrow{\text{ID}}$ vice versa. On the other hand, with the labels "Cub" and "Lin" we refer to the cubic and the linear radial correction models, respectively; see (20). In all reported cases, the early stopping strategy from Sect. IV-E is imposed.

All the tests were run on a server with the following characteristics: 16 Cores, Processors 2xXeon E5-2667v3 3.2 GHz and RAM 192 GB; our implementation requires at most 5GB of RAM. In experimentally comparing the variants of our method with the algorithms proposed in [8] and [9], we noticed that [8] was tested on images of size 3000×4000 that are, on average, larger than the in-camera corrected images in our dataset (refer to Table II for the image sizes in each subset). This explains the slightly worse performance measured here (with downsampling) compared to that reported in [8].

In Table I we provide the fixed thresholds τ_c (measured over the entire test dataset) that ensure False Positive Rates (FPR) of 0.05 and 0.01 together with the corresponding True Positive Rates (TPR) for the different variants of our method and those in [8] and [9] (with and without DS).

A breakdown by subset of the previous table is given in Table II which also shows the time consumed to declare a match (under H_1) by the different alternatives (with early stopping in those cases where it applies).[5] For reasons of space, we have excluded the method in [9] which yields a modest performance, as well as the worst-performing variants of our method (cf. Table I); see [21] for fully comprehensive results. The Receiver Operating Characteristic (ROC) curves for the variants in Table II are plotted in Fig. 8, where we have also added for comparison the baseline (BL) obtained by using $\text{PCE}(\hat{\mathbf{K}}', \mathbf{W})$ (i.e., with no transformations of either the PRNU or the residual) as test statistic.

[5]The time consumed when no early stopping is in force is given in the technical report [21].

TABLE I

THRESHOLDS REQUIRED TO ACHIEVE FPR=0.05 AND FPR=0.01 AND CORRESPONDING TPRS FOR DIFFERENT METHODS AND VARIANTS

	$\tau_{0.05}$	$\tau_{0.01}$	$\text{TPR}_{0.05}$	$\text{TPR}_{0.01}$
$\overrightarrow{\text{ID}}$, Lin	98.86	112.71	0.96	0.94
$\overrightarrow{\text{DI}}$, Lin	90.01	105.63	0.97	0.95
$\overrightarrow{\text{ID}}$, Cub	73.48	90.28	0.99	0.98
$\overrightarrow{\text{DI}}$, Cub	71.13	84.75	0.99	0.98
Inv, Lin	97.66	111.72	0.93	0.91
Dir, Lin	90.01	105.63	0.95	0.92
Inv, Cub	73.48	90.29	0.98	0.98
Dir, Cub	71.13	84.57	0.99	0.98
2W, Cub	71.12	84.34	0.99	0.99
[8]	4.81	8.36	0.86	0.82
[8] no DS	7.65	10.54	0.94	0.93
[9]	2.83	5.50	0.78	0.73

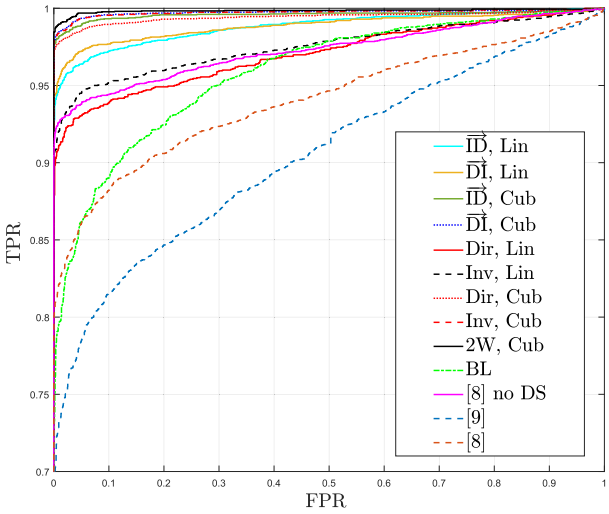

Fig. 8. ROCs obtained with the variants of our method, [8] (with and without DS) and [9] on the test dataset.

From the results in Fig. 8 and Table I, it is possible to conclude that the best performing variants of our method correspond to the cubic correction model ("Cub"), with "$\overrightarrow{\text{DI}}$", "Dir", "Inv" and "2W" all achieving similar TPRs for the target FPRs. On the other hand, the average execution time of the "one way" variants, i.e. "Dir" and "Inv", is lower because only one statistic has to be computed per iteration. The experiments, conducted on a dataset composed of a variety of radial corrections, show that our variants outperform [8] (both with or without DS) in terms of TPR. Moreover, thanks to our early stopping strategy, our fastest variants (i.e., "Dir, Cub" and "Inv, Cub") achieve under H_1 execution times that are comparable to [8] with DS. We also note that the original solution proposed in [8] (i.e. with DS) achieves a limited performance both on low-resolution devices (i.e. SX230 and ZS7) and in presence of complex out-camera radial corrections as those applied by Adobe Lightroom. Adapting [8] to avoid DS results in a significant performance increase in those difficult cases, at the expense of a much more costly execution. Nevertheless, for some severe radial corrections like those in

TABLE II

TPRS OF THE DIFFERENT VARIANTS OF OUR METHOD AND [8] (WITH AND WITHOUT DS), ANDTHE AVERAGE TIME CONSUMED TO DECLARE A MATCH FOR SPECIFIC DEVICES/SOFTWARE. IN **BOLD** WE HIGHLIGHT THE FASTEST AND MOST ACCURATE VARIANTS OF OUR METHOD

	$\overrightarrow{\text{ID}}$, Lin		$\overrightarrow{\text{DI}}$, Lin		$\overrightarrow{\text{ID}}$, Cub		$\overrightarrow{\text{DI}}$, Cub		Inv, Cub		Dir, Cub		2W, Cub		[8] no DS		[8]	
	TPR	time	TPR	time	TPR	time	TPR	time	TPR	time	TPR	time	TPR	time	TPR	time	TPR	time
GIMP [3456 × 5184]	0.98	220.1	0.98	303.6	0.98	193.9	**0.99**	204.8	0.98	**182**	0.98	196.2	**0.99**	326.4	0.97	962.3	0.97	85.3
LIGHTROOM [3456 × 5184]	0.93	**237.8**	0.95	330.3	0.94	288.4	0.97	309.1	0.94	274.6	0.97	295.1	**0.98**	467.4	0.6	930.8	0.44	87.9
LIGHTROOM* [3456 × 5184]	0.95	137.5	0.92	222.6	**0.99**	158.8	0.98	**126.5**	**0.99**	127.7	0.98	129.7	0.98	220.5	0.51	930.9	0.4	91.6
PHOTOSHOP [3456 × 5184]	0.97	**151.2**	**0.98**	159.5	0.97	165.4	0.97	176.2	0.97	170.7	0.97	167.3	0.97	293	0.96	974.6	0.91	107.2
PT LENS [3456 × 5184]	0.97	**181.3**	0.99	247.3	0.98	248.9	0.99	253.6	0.98	256.3	**1**	238.2	**1**	412.4	0.91	918.3	0.8	100.3
S9100 [3000 × 4000]	0.99	114.2	0.98	117.3	0.99	**72.9**	0.99	99.8	0.99	92.9	**1**	92.5	**1**	160.3	1	598.3	0.98	81.7
SX210 [3240 × 4320]	0.96	256.8	0.97	262.5	1	127.5	1	139.1	**1**	**121.8**	**1**	126.8	**1**	210.3	1	723.4	0.99	81.3
SX230 [1584 × 2816]	0.87	33.8	0.9	29	1	**23**	1	24.2	**0.98**	41	**0.98**	41.8	**0.98**	70.5	0.98	229.0	0.83	25.1
SX40 [2664 × 4000]	1	105.1	0.96	145	1	93.4	1	103.4	**1**	**88.6**	**1**	93.6	**1**	154	1	526.3	0.98	56.3
ZS7 [1920 × 2560]	0.76	42.7	0.81	35	0.98	**27.7**	0.98	29.8	0.93	50.5	**0.99**	46.9	**0.99**	79.6	0.98	255.2	0.79	27.8

our "Lightroom*" subset, using the full resolution in [8] is still not sufficient. In contrast, our method is able to adapt to this high complexity and offers an excellent performance with an affordable execution time.

An analysis of the results for our variants across the different subsets reveals that both out-camera radial distortion corrections and small images represent the most difficult scenarios. As already mentioned, complex corrections probe the limits of the expressive capacity of the radial transformations that need to be applied; on the other hand, small images have a lower PCE (or CPCE) thus making it more difficult to score above the threshold under H1. In this sense, processing pipelines that further degrade the signal strength (e.g. compressions applied by social networks) become great challenges that deserve future attention. Concerning complex radial distortion corrections, preliminary experiments carried out by replacing in our variants the proposed local search algorithm by a much more time-consuming exhaustive search reveal that there is a significant margin of improvement in the former.

Considering the trade-off between performance and speed, our recommended variant is "Dir, Cub", which is the one that we used in the other experiments discussed in the following sections.

B. Impact of Annuli's Width

Next, we present results supporting the choice for the width of the annuli Δ_k. Recall that in this paper we assume that $\Delta_k = \Delta$ except for the inner disk. Fig. 9 plots the ROC curves for the previous dataset and different values of $\Delta \cdot D_2$. The "Dir, Cub" variant of our method was selected. The legend shows the Area Under the Curve (AUC) obtained by integrating the ROC for FPRs in the interval $[0, 0.05)$ and normalizing it by 0.05, so that for a perfect detector this AUC would be 1; this quantity is denoted as AUC@0.05. As we can see, when $\Delta \cdot D_2$ is too large, the expressive capacity of the third-order mapping is less adequate to correctly approximate the true radial distortion correction in the most complex

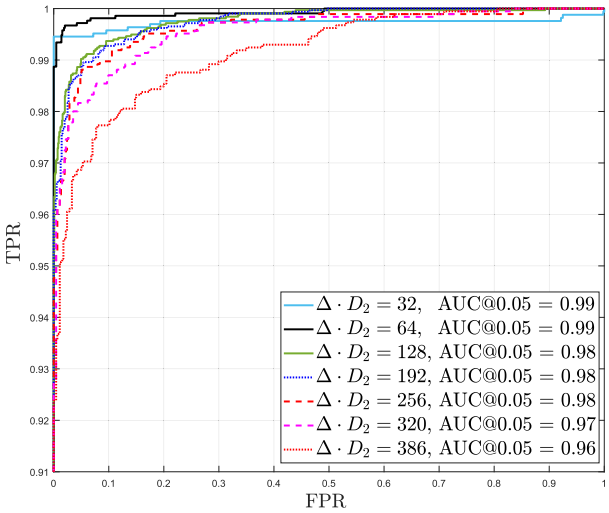

Fig. 9. ROC curves for different values of annulus width $\Delta \cdot D_2$.

cases. On the other hand, when $\Delta \cdot D_2$ is too small, the signal to noise ratio may be insufficient for our adaptive method to work properly. In any case, our results show that performance is quite robust to the choice of $\Delta \cdot D_2$. Thus, our choice of $\Delta \cdot D_2 = 64$ was motivated by it achieving the best performance and the fact that computational complexity decreases with $\Delta \cdot D_2$ (e.g. the execution time for $\Delta \cdot D_2 = 64$ is 0.61 times that for $\Delta \cdot D_2 = 32$ on average across the dataset).

C. Verification of the Approximation in (35)

In Fig. 10 we visualize the goodness of the approximation presented in (35), for $T_{\boldsymbol{\alpha}^*,c}$, that is, the "Dir, Cub" variant. For this experiment, we took 20 random images for each of the following models in the dataset of Sect. V-A: Canon SX230, SX210, Panasonic ZS7, Nikon S9100 (all corrected in-camera) and the Canon1200D (corrected with Adobe Lightroom). In order to apply $T_{\boldsymbol{\alpha}^*,c}$ on the left hand side of (35) we did so in annulus-by-annulus basis, that is, deriving the transformed

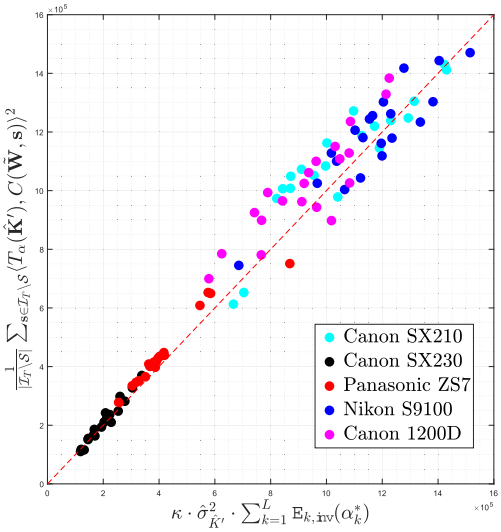

Fig. 10. Scatter-plot of the two sides of (35) computed for images from different camera models in the dataset.

coordinates $T_{\alpha_k^*,c}$ for each annulus and then computing the transformed fingerprint $T_{\boldsymbol{\alpha}^*,c}(\hat{\mathbf{K}}')$ by interpolating $\hat{\mathbf{K}}$ at those coordinates. Fig. 10 represents a scatter-plot in which the coordinates of every point are obtained by computing the right and left hand sides of (35). As we see, the approximation that we proposed in the paper is reasonable (even more so considering the interpolation carried out to practically compute the left hand side of (35)). We can also see that for larger images the approximation is noisier; this can be attributed to the larger variance of the sample variance estimator given by the denominator of the PCE, as discussed in [21]. The absolute relative error for the samples considered in Fig. 10 has a mean value of 0.069.

D. Experiments With Smartphones From [10]

For the second set of experiments, we decided to assess the presence of radial distortion corrections on common smartphones, and analyze in this context the performance of some of the methods discussed in this paper when trying to solve the mismatches due to those corrections. To this end, we decided to use a subset of the database recently published by Iuliani et al. [10] which contains 33,000 Flickr images belonging to 45 smartphones and 25 DSLR camera models with several different devices for each model (the number of test images and devices per model are indicated in the first two columns of table III). Due to the fact that, as it will be explained below, one of the tested approaches required a brute-force search, we had to limit the set of tested models to the ones shown in Table III, which were randomly picked from the original dataset. For those selected models, we used all the available images. The results on the full dataset will be reported elsewhere. The fingerprints were computed for each device of each model using 35 images (those provided for the reference fingerprint by the authors of [10]) that were later discarded for testing.

Specifically, in this experiment, noticing that the results in [10] contain a large number of false negatives, we wanted

TABLE III
STANDARD TRUE POSITIVE RATES AND NEW DISCOVERY RATES FOR CAMERA MODELS IN THE SECOND DATASET

	No. Dev.	No. Img.	STPR	Ours NDR	[8] NDR	Exhaust. [8] NDR
iphone11pro	7	124	0.77	0.02	0.09	0
iphone11promax	9	232	0.82	0.02	0.09	0
iphone6	10	341	0.76	0.05	0.01	0.04
iphone7	10	295	0.71	0.08	0.04	0
huawei ane lx1	10	336	0.68	**0.33**	0.03	0.08
huawei sne lx1	10	346	0.58	**0.70**	0.05	0.05
motoe5play	9	293	0.65	**0.20**	0.01	0.06
808pureview	8	280	1	-	-	-
oneplusa6013	8	272	0.61	**0.25**	0.02	0.07
oppoa92020	3	103	0.61	0.02	0.04	0
realmec2	2	67	0.93	0	0	0

to find out whether any of those could be attributed to in-camera radial distortion corrections and thus be reverted by the algorithms discussed in this paper. Thus, we chose the method "Dir, Cub" as representative of our variants, and [8] for the state of the art. Furthermore, to decouple the limitations of the search method proposed in [8] from the expressive constraints of the family of third-order radial transformations considered therein and given by (10), we decided to modify the method in [8] so that parameter α is found through an exhaustive search with step size 0.01 and range $[-0.3, 0.3]$.

Next, we explain the quantities reflected in the table columns. For every given camera model, the *standard true positive rate* (STPR) is the fraction of images that are correctly matched using the standard PCE (Eq. (5)) without any radial transformation. The detection threshold is set for a FPR=0.01 separately for each model[6] (i.e. H_0 is constructed with all images taken with devices from the same model other than the device under test); this is advisable in the current experiment because of: 1) the availability of many exemplars of the same model, which allows us to construct a rich null hypothesis; 2) the fact that for some models we have observed systematic biases on the PCE values under both H_0 and H_1 (for instance, for the Nokia Pure View 808 model a STPR=1 is prominently achieved when the threshold is set using only images from the same model, different device, to compose H_0). The *new discovery rate* (NDR) is the fraction of false negatives of the standard PCE test that can be matched after applying the algorithm under study. Again, the threshold is set specifically for each model, which requires that images from H_0 are also subjected to the algorithm under study to properly evaluate the FPR. To give an example, suppose that for a model we have 100 test images, 70 of which are correctly detected with the standard PCE test, so STPR=0.7; if for the remaining 30 false negatives, 10 of them are correctly detected after applying a distortion inversion algorithm (i.e, there were 10 newly discovered positives), then NDR=10/30=0.33.

From Table III we can see two types of cases: 1) for the iPhones and the Oppo A9 2020 the NDRs are relatively small (but not zero), so it cannot be entirely ruled out that the new discoveries are just false positives. 2) For the two Huawei models, the Moto e5 Play, and the One Plus 6T our

[6]While for other experiments in the paper we set FPR=0.05, here we chose a more conservative threshold in order to achieve larger statistical significance.

method achieves significantly many more new discoveries than those of the two other approaches, including the exhaustive search. The new discoveries made by our method range from 20% to 70%, which is a remarkable result, as it leads to a dramatic reduction in the false negatives reported in [10]. The statistical significance of the new discoveries is supported by their p-values.[7] Let μ_p and σ_p respectively denote the mean and standard deviation (std) of the p-values corresponding to the new discoveries. For those new discoveries made by our method their respective means and stds of p-values are as follows: Huawei P20 lite: $\mu_p = 2.1 \cdot 10^{-3}$, $\sigma_p = 2 \cdot 10^{-3}$; Huawei mate 20 lite: $\mu_p = 1.3 \cdot 10^{-3}$, $\sigma_p = 3 \cdot 10^{-3}$; Moto e5 Play: $\mu_p = 1.7 \cdot 10^{-3}, \sigma_p = 2 \cdot 10^{-3}$; One Plus 6T: $\mu_p = 2 \cdot 10^{-4}$, $\sigma_p = 1 \cdot 10^{-3}$. On the other hand, failure of [8] to make new discoveries in these cases, even with brute-force search, is a clear sign that for those camera models the radial corrections applied in-camera are more complex than the third-order model described in (10), and thus require a more flexible approach, as afforded by our proposed methods.

VI. Conclusion

In this paper, we have proposed an adaptive method for PRNU-based camera attribution that is able to cope with complex radial distortion corrections, as those performed in-camera by most compact models and out-camera by image processing software. Existing approaches try to either "correct" the reference fingerprint or invert the correction by applying a further geometric transformation that, in order to avoid a combinatorial explosion, must use a reduced number of parameters. In turn, this limitation accounts for unsatisfactory performance when complex radial distortion corrections are in effect, an undesirable aspect in view of the trend of more elaborate transformations that are made possible by ever more powerful distortion correction firmware/software. Our approach is radically different: by applying a divide-and-conquer principle, embodied in the use of annuli, we are able to: 1) allow for complex distortion corrections, as locally the transformation undergone by each annulus is much simpler; 2) implement an early stopping strategy that offers large computational savings. The results presented in the paper clearly reveal that our algorithm (in most of its variants) outperforms the state of the art when accuracy and computational load are considered.

We believe that the adaptive approach proposed here could also be fruitful in other very challenging camera attribution scenarios with a number of latent parameters, such as in HDR images [7], in-camera-stabilized videos [25], and emerging in-camera processing [10].

Experiments carried out with complex radial correction distortions show that a local exhaustive search significantly outperforms our prediction-based algorithm; moreover, for those outer annuli that get out of the image boundaries, the lower signal to noise ratio often yields estimation errors. Thus, the design of more effective, yet computationally affordable, local search algorithms is an open problem. On the other hand, our methods assume that the optical center of the radial transformation is known; when this is not the case (e.g. due to image cropping), the number of unknown parameters grows, as does the need for efficient search algorithms.

[7]Note that, by selection of the threshold, all p-values are $\leq 10^{-2}$.

Appendix A
Derivation of the Estimator of α_k^*

In this Appendix we derive a plausible estimator of α_k^* under the inverse approach; the derivation would be identical for the direct approach and, hence, is skipped here. See the definition of α_k^* in (26). We introduce a super-index in α_k to enumerate the elements of the candidate set $\mathcal{A}_k$, i.e., $\{\alpha_k^{(n)} : n = 1, \cdots, A_k\} = \mathcal{A}_k$.

We assume that H_1 holds, i.e., $\mathbf{I}$ contains $\mathbf{K}'$, and the following model for the residuals:

$$[T^{-1}_{\alpha_k^{(n)}}(\tilde{\mathbf{W}})]_{i,j} = \gamma^{(n)}_{i,j}[T^{-1}_{\alpha_k^{(n)}}(T_{\alpha_k^\dagger}(\hat{\mathbf{K}}'))]_{i,j} + N^{(n)}_{i,j} \tag{38}$$

for all $(i, j) \in \mathcal{Q}_{k,\mathsf{inv}}(\alpha_k^{(n)})$ and $n \in \{1, \cdots, A_k\}$. In (38) $\alpha_k^\dagger$ represents the *true* (locally for the kth annulus) value of α. The multipliers $\gamma^{(n)}_{i,j}$ are non-negative and take into account both the multiplicative effect of the image $\mathbf{I}$ and the gain of the effective denoising filter (which also impacts on the estimate $\hat{\mathbf{K}}'$ of the true PRNU). We argue that these multipliers are very hard to estimate accurately; as a consequence, a full maximum likelihood decision will not be possible and some simplifications will be required. One such simplification is to consider that the cross-correlations between $\hat{\mathbf{K}}'$ and $T^{-1}_{\alpha_k^{(n)}}(\tilde{\mathbf{W}})$, for all $n = 1, \cdots, A_k$, constitute a set of sufficient statistics for the estimation problem. Recall from (24) that these cross-correlations are denoted by $\Phi_{k,\mathsf{inv}}(\alpha_k^{(n)})$.

We make the following hypotheses:

1) *Spikiness:* The $\alpha_k^{(n)}$ are sufficiently separated so that the $\Phi_{k,\mathsf{inv}}(\alpha_k^{(n)})$ are mutually uncorrelated and $\mathbb{E}\{\Phi_{k,\mathsf{inv}}(\alpha_k^{(n)})\} = 0$, for all $n = 1, \cdots, A_k$, except for $n = l$, where l is such that $\alpha_k^{(l)}$ is the closest to the true value $\alpha_k^\dagger$ and the expectation is taken over the underlying distribution of $\mathbf{K}'$. This hypothesis is reasonable in view of the spikiness of the PCE with α (see Fig. 2). We also assume that $\alpha_k^{(l)}$ is close enough to $\alpha_k^\dagger$ so that $[T^{-1}_{\alpha_k^{(l)}}(T_{\alpha_k^\dagger}(\hat{\mathbf{K}}'))]_{i,j} \approx \hat{K}'_{i,j}$ for all $(i, j) \in \mathcal{Q}_{k,\mathsf{inv}}(\alpha_k^{(l)})$.

2) *Uncorrelatedness:* In (38), $N^{(n)}_{i,j}$ and $\gamma^{(n)}_{i,j}[T^{-1}_{\alpha_k^{(l)}}(\hat{\mathbf{K}}')]_{i,j}$ are zero-mean and mutually uncorrelated for all $(i, j) \in \mathcal{Q}_{k,\mathsf{inv}}(\alpha_k^{(n)})$ and $n \in \{1, \cdots, A_k\}$. For any $l, n \in \{1, \cdots, A_k\}$, $l \neq n$, the variables $\hat{K}'_{i,j} \cdot N^{(n)}_{i,j}$ and $\hat{K}'_{u,v} \cdot N^{(l)}_{u,v}$ are mutually uncorrelated for every $(i, j) \in \mathcal{Q}_{k,\mathsf{inv}}(\alpha_k^{(n)})$ and every $(u, v) \in \mathcal{Q}_{k,\mathsf{inv}}(\alpha_k^{(l)})$.

3) *Weak PRNU:* In (38), $\left|\gamma^{(n)}_{i,j}[T^{-1}_{\alpha_k^{(n)}}(T_{\alpha_k^\dagger}(\hat{\mathbf{K}}'))]_{i,j}\right| \ll |N_{i,j}|$ for a large number of pixels of each annulus; we write this more precisely as

$$\sum_{(i,j)\in\mathcal{Q}_{k,\mathsf{inv}}(\alpha_k^{(n)})} \left(\gamma^{(n)}_{i,j}\right)^2 [T^{-1}_{\alpha_k^{(n)}}(T_{\alpha_k^\dagger}(\hat{\mathbf{K}}'))]^2_{i,j} \ll \sum_{(i,j)\in\mathcal{Q}_{k,\mathsf{inv}}(\alpha_k^{(n)})} N^2_{i,j} \tag{39}$$

for all $n \in \{1, \cdots, A_k\}$.

As a consequence of the spikiness and uncorrelatedness assumptions above and the Central Limit Theorem (which

is applicable if we assume that $|\mathcal{Q}_{k,\mathsf{inv}}(\alpha_k^{(n)})|$ is large for all $n \in \{1, \cdots, A_k\}$), the variables $\Phi_{k,\mathsf{inv}}(\alpha_k^{(n)})$ will be well modeled by independent Gaussian distributions, so the cross-correlations will be

$$\Phi_{k,\mathsf{inv}}(\alpha_k^{(n)}) \sim \mathcal{N}(0, (\sigma^{(n)})^2), \quad n = 1, \cdots, A_k, n \neq l \tag{40}$$

$$\Phi_{k,\mathsf{inv}}(\alpha_k^{(l)}) \sim \mathcal{N}(\mu^{(l)}, (\sigma^{(l)})^2) \tag{41}$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes a Gaussian with mean μ and variance σ^2, $\mu^{(l)}$ denotes the expected value of the cross-correlation for the value of $\alpha_k^{(n)} \in \mathcal{A}_k$ that is closest to $\alpha_k^{\dagger}$, and $(\sigma^{(n)})^2$, $n = 1, \cdots, A_k$, denote the variances of the cross-correlations.

For all $n = 1, \cdots, A_k$, the variances $(\sigma^{(n)})^2$ can be written as $\mathrm{Var}\{\sum_{(i,j)\in\mathcal{Q}_{k,\mathsf{inv}}(\alpha_k)} \hat{K}'_{i,j} \cdot N_{i,j}\} \approx \hat{\sigma}^2_{\hat{K}'} \sum_{(i,j)\in\mathcal{Q}_{k,\mathsf{inv}}(\alpha_k)} N^2_{i,j}$. As a consequence of the weak PRNU assumption $\sum_{(i,j)\in\mathcal{Q}_{k,\mathsf{inv}}(\alpha_k)} N^2_{i,j} \approx \mathsf{E}_{k,\mathsf{inv}}(\alpha_k^{(n)})$. Therefore, $(\sigma^{(n)})^2 \approx \hat{\sigma}^2_{\hat{K}'} \mathsf{E}_{k,\mathsf{inv}}(\alpha_k^{(n)})$.

Let $f_{\mathcal{N}}(Y; \mu, \sigma)$ denote the Gaussian pdf on random variable $Y \sim \mathcal{N}(\mu, \sigma^2)$. Also, let $\mathcal{E}_l$ denote the event "$\alpha_k^{(l)}$, $l \in \{1, \cdots, A_k\}$ is the closest to the true value $\alpha_k^{\dagger}$". Then the likelihood of jointly observing the cross-correlations $\Phi_{k,\mathsf{inv}}(\alpha_k^{(n)})$ conditioned on $\mathcal{E}_l$ is

$$\begin{aligned} &f(\Phi_{k,\mathsf{inv}}(\alpha_k^{(1)}), \cdots, \Phi_{k,\mathsf{inv}}(\alpha_k^{(A_k)})|\mathcal{E}_l) \\ &= f_{\mathcal{N}}(\Phi_{k,\mathsf{inv}}(\alpha_k^{(l)}); \mu^{(l)}, \sigma^{(l)}) \cdot \prod_{\substack{n=1 \\ n\neq l}}^{A_k} f_{\mathcal{N}}(\Phi_{k,\mathsf{inv}}(\alpha_k^{(l)}); 0, \sigma^{(n)}) \end{aligned} \tag{42}$$

The maximum likelihood estimator would be obtained by maximizing the likelihood in (42) with respect to l. The estimator will not change if we divide (42) by $\prod_{n=1}^{A_k} f_{\mathcal{N}}(\Phi_{k,\mathsf{inv}}(\alpha_k^{(n)}); 0, \sigma^{(n)})$; this gives the following simpler likelihood function

$$\begin{aligned} &L(\Phi_{k,\mathsf{inv}}(\alpha_k^{(1)}), \cdots, \Phi_{k,\mathsf{inv}}(\alpha_k^{(A_k)})|\mathcal{E}_l) \\ &= \frac{f_{\mathcal{N}}(\Phi_{k,\mathsf{inv}}(\alpha_k^{(l)}); \mu^{(l)}, \sigma^{(l)})}{f_{\mathcal{N}}(\Phi_{k,\mathsf{inv}}(\alpha_k^{(l)}); 0, \sigma^{(l)})} \end{aligned} \tag{43}$$

Taking the logarithm and simplifying, we find that the maximum likelihood estimator is equivalent to solving

$$l^* = \arg \max_{l=1,\cdots,A_k} \psi^{(l)} \tag{44}$$

where

$$\psi^{(l)} \doteq \frac{\mu^{(l)} \cdot \Phi_{k,\mathsf{inv}}(\alpha_k^{(l)})}{(\sigma^{(l)})^2} - \frac{1}{2}\frac{(\mu^{(l)})^2}{(\sigma^{(l)})^2} \tag{45}$$

and making $\alpha_k^* = \alpha_k^{(l^*)}$.

Notice that, as discussed above, $(\sigma^{(l)})^2$ can be replaced by its estimator $\hat{\sigma}^2_{\hat{K}'} \mathsf{E}_{k,\mathsf{inv}}(\alpha_k^{(l)})$ in (45). Unfortunately, producing a reliable estimator of $\mu^{(l)}$ is not feasible due to the unavailability of the gains $\gamma_{i,j}^{(l)}$. For this reason, we turn our attention to suboptimal estimators that can be practically implemented. If we assume that for all l in $\{1, \cdots, A_k\}$ both $\mu^{(l)}$ and the ratio $\mu^{(l)}/\sigma^{(l)}$ do not vary significantly around their respective means, we can think of replacing $\mu^{(l)}$ and $\mu^{(l)}/\sigma^{(l)}$ in (45) by those means. This yields the simplified functional

$$\psi'^{(l)} \doteq \Phi_{k,\mathsf{inv}}(\alpha_k^{(l)})/(\sigma^{(l)})^2 \tag{46}$$

to be used in (44). After replacing $(\sigma^{(l)})^2$ in (46) by its estimator $\hat{\sigma}^2_{\hat{K}'} \mathsf{E}_{k,\mathsf{inv}}(\alpha_k^{(l)})$, and dropping $\hat{\sigma}^2_{\hat{K}'}$ because it is independent of l, we obtain the proposed (27).

It is interesting to evaluate the loss of performance that results when using (46) instead of (45). We do so by assuming w.l.o.g. that $\mathcal{E}_l$ holds and estimate the probabilities that a given $n \in \{1, \cdots, A_k\}$, $n \neq l$, produces a larger value than for $n = l$ in $\psi^{(n)}$ and $\psi'^{(n)}$. Then, we compare the two resulting probabilities in terms of the effective signal-to-noise ratios (SNR). Therefore, in this case, following (40), we have that for $n \neq l$, $\Phi_{n,\mathsf{inv}}(\alpha_k^{(n)}) \sim \mathcal{N}(0, (\sigma^{(n)})^2)$, and $\Phi_{k,\mathsf{inv}}(\alpha_k^{(l)}) \sim \mathcal{N}(\mu^{(l)}, (\sigma^{(l)})^2)$. Thus, when $\mathcal{E}_l$ holds, $\psi^{(l)} \sim \mathcal{N}\left((\mu^{(l)})^2/(\sqrt{2}\sigma^{(l)})^2, (\mu^{(l)})^2/(\sigma^{(l)})^2\right)$ and $\psi^{(n)} \sim \mathcal{N}\left(-(\mu^{(n)})^2/(\sqrt{2}\sigma^{(n)})^2, (\mu^{(n)})^2/(\sigma^{(n)})^2\right)$, $n \neq l$. Since $\psi^{(l)}$ and $\psi^{(n)}$ are independent, the probability that $\psi^{(n)} \geq \psi^{(l)}$ when $\mathcal{E}_l$ holds is the probability that the random variable $\psi^{(l)} - \psi^{(n)}$ is less than zero. And since both variables are Gaussian, so is their difference. Therefore, $\psi^{(l)} - \psi^{(n)} \sim \mathcal{N}(\omega_{n,l}/2, \omega_{n,l})$, where

$$\omega_{n,l} \doteq \frac{(\mu^{(l)})^2}{(\sigma^{(l)})^2} + \frac{(\mu^{(n)})^2}{(\sigma^{(n)})^2} \tag{47}$$

If we define the effective SNR as the ratio between the squared mean and the variance of $\psi^{(l)} - \psi^{(n)}$, then we find that $\mathrm{SNR}_{\psi} = \omega_{n,l}/4$, where the subindex ψ indicates that we are using the estimator in (45).

For the simplified estimator in (46), a similar derivation leads to showing that

$$\psi'^{(l)} - \psi'^{(n)} \sim \mathcal{N}\left(\frac{\mu^{(l)}}{(\sigma^{(l)})^2}, \left[\frac{1}{(\sigma^{(l)})^2} + \frac{1}{(\sigma^{(n)})^2}\right]\right) \tag{48}$$

for which the effective SNR, denoted as $\mathrm{SNR}_{\psi'}$ is now

$$\mathrm{SNR}_{\psi'} = \frac{(\mu^{(l)})^2/(\sigma^{(l)})^2}{\frac{(\sigma^{(l)})^2}{(\sigma^{(n)})^2} + 1} \tag{49}$$

In order to compare the effective SNRs, we compute their ratio:

$$\frac{\mathrm{SNR}_{\psi}}{\mathrm{SNR}_{\psi'}} = \frac{1 + \left(\frac{\mu^{(n)}}{\mu^{(l)}}\right)^2 \cdot \left(\frac{\sigma^{(l)}}{\sigma^{(n)}}\right)^2}{4} \cdot \left(\left(\frac{\sigma^{(l)}}{\sigma^{(n)}}\right)^2 + 1\right) \tag{50}$$

To get a cleaner interpretation of this result, we can further assume that both $\mu^{(n)}$ and $(\sigma^{(n)})^2$ are proportional to the cardinality of the support set $|\mathcal{Q}_{k,\mathsf{inv}}(\alpha_k^{(n)})|$. This way, if we let $\beta_{n,l} \doteq |\mathcal{Q}_{k,\mathsf{inv}}(\alpha_k^{(n)})|/|\mathcal{Q}_{k,\mathsf{inv}}(\alpha_k^{(l)})|$, we can write that $\mu^{(n)}/\mu^{(l)} = \beta_{n,l}$ and $(\sigma^{(l)})^2/(\sigma^{(n)})^2 = \beta_{n,l}^{-1}$. Then, substituting into (50) we find that

$$\frac{\mathrm{SNR}_{\psi}}{\mathrm{SNR}_{\psi'}} = \frac{(1+\beta_{n,l})^2}{4\beta_{n,l}} = 1 + \frac{(1-\beta_{n,l})^2}{4\beta_{n,l}} \tag{51}$$

which is clearly larger than one for all $\beta_{n,l} \geq 0$, $\beta_{n,l} \neq 1$. This confirms that, as expected, for any $\beta_{n,l} \neq 1$ there is a loss of effective SNR with respect to the optimal estimator. However, in practice this loss will be rather small: for instance, suppose that $|\mathcal{Q}_{k,\mathsf{inv}}(\alpha_k^{(n)})|$ is within the range of 20% larger

and 20% smaller than $|\mathcal{Q}_{k,\mathsf{inv}}(\alpha_k^{(l)})|$, then the effective SNR for the suboptimal detector is at most 0.054 dB smaller than the corresponding to the optimal one.

This supports the plausibility of the proposed simplified detector.

ACKNOWLEDGMENT

Funding for open access charge: Universidade de Vigo/CISUG.

REFERENCES

[1] J. Luka, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 205–214, Jun. 2006.

[2] K. Rosenfeld and H. T. Sencar, "A study of the robustness of PRNU-based camera identification," *Proc. SPIE*, vol. 7254, Feb. 2009, Art. no. 72540M.

[3] S. Taspinar, M. Mohanty, and N. Memon, "Camera fingerprint extraction via spatial domain averaged frames," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3270–3282, 2020.

[4] P. Korus and J. Huang, "Multi-scale analysis strategies in PRNU-based tampering localization," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 4, pp. 809–824, Apr. 2017.

[5] M. Goljan, "Digital camera identification from images—Estimating false acceptance probability," in *Proc. Int. Workshop Digital Watermarking*. Cham, Switzerland: Springer, 2008, pp. 454–468.

[6] S. Taspinar, M. Mohanty, and N. Memon, "Source camera attribution using stabilized video," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2016, pp. 1–6.

[7] M. Darvish Morshedi Hosseini and M. Goljan, "Camera identification from HDR images," in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, Jul. 2019, pp. 69–76.

[8] M. Goljan and J. Fridrich, "Sensor-fingerprint based identification of images corrected for lens distortion," *Proc. SPIE*, vol. 8303, Feb. 2012, Art. no. 83030H.

[9] M. Goljan and J. Fridrich, "Estimation of lens distortion correction from single images," *Proc. SPIE*, vol. 9028, Jan. 2014, Art. no. 90280N.

[10] M. Iuliani, M. Fontani, and A. Piva, "A leak in PRNU based source identification—Questioning fingerprint uniqueness," *IEEE Access*, vol. 9, pp. 52455–52463, 2021.

[11] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, "Determining image origin and integrity using sensor noise," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 1, pp. 74–90, Mar. 2008.

[12] M. Goljan, J. Fridrich, and T. Filler, "Large scale test of sensor fingerprint camera identification," *Proc. SPIE*, vol. 7254, Feb. 2009, Art. no. 72540I.

[13] M. K. Mihcak, I. Kozintsev, K. Ramchandran, and P. Moulin, "Low-complexity image denoising based on statistical modeling of wavelet coefficients," *IEEE Signal Process. Lett.*, vol. 6, no. 12, pp. 300–303, Dec. 1999.

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

[15] W. Hugemann, "Correcting lens distortions in digital photographs," in *Proc. 19th Eur. Assoc. Accident Res. Anal. (EVU) Conf.*, Prague, Czech Republic, 2010. [Online]. Available: https://www.evu-online.org/correcting-lens-distortions-in-digital-photographs

[16] H. Li and R. Hartley, "A non-iterative method for correcting lens distortion from nine point correspondences," *OMNIVIS*, vol. 2, no. 7, pp. 1–8, 2005.

[17] J. Perš and S. Kovacic, "Nonparametric, model-based radial lens distortion correction using tilted camera assumption," in *Proc. Comput. Vis. Winter Workshop*, vol. 1, 2002, p. 286.

[18] P. Drap and J. Lefèvre, "An exact formula for calculating inverse radial lens distortions," *Sensors*, vol. 16, no. 6, p. 807, Jun. 2016.

[19] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables*. New York, NY, USA: Dover, 1964.

[20] A. Montibeller and F. Pérez-González, "Exploiting PRNU and linear patterns in forensic camera attribution under complex lens distortion correction," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[21] A. Montibeller and F. Pérez-González, "Technical report additional material for, 'an adaptive method for camera attribution under complex radial distortion corrections,"' DISI, Univ. Trento, Italy, Tech. Rep., 2022, doi: 10.13140/RG.2.2.20038.96323/1.

[22] B. Widrow and S. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1985.

[23] F. Pérez-González, M. Masciopinto, I. González-Iglesias, and P. Comesaña, "Fast sequential forensic detection of camera fingerprint," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3902–3906.

[24] M. Goljan, "Blind detection of image rotation and angle estimation," *Electron. Imag.*, vol. 30, no. 7, p. 158, Jan. 2018.

[25] S. Mandelli, P. Bestagini, L. Verdoliva, and S. Tubaro, "Facing device attribution problem for stabilized video sequences," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 14–27, 2020.

Andrea Montibeller (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in information and communications engineering in 2018 and 2020, respectively. He is currently pursuing the Ph.D. degree in information engineering and computer science with the University of Trento, Italy.

In 2023, he was a Research Scholar with the Department of Electrical and Computer Engineering, Drexel University, Philadelphia. In 2018, he was a Visiting Researcher with the Department of Signal Theory and Communications, University of Vigo, Spain. His research interests are AI generated image and video forgery detection and localization and camera source attribution.

Fernando Pérez-González (Fellow, IEEE) received the degree in telecommunication engineering from the University of Santiago, Santiago, Spain, in 1990, and the Ph.D. degree in telecommunications engineering from the University of Vigo, Vigo, Spain, in 1993.

He is currently a Professor with the School of Telecommunication Engineering, University of Vigo. From 2007 to 2010, he was the Program Manager of the Spanish National Research and Development Plan on Electronic and Communication Technologies, Ministry of Science and Innovation. From 2009 to 2011, he was the Prince of the Asturias Endowed Chair of Information Science and Technology, The University of New Mexico, Albuquerque, NM, USA. From 2007 to 2014, he was the Executive Director of the Galician Research and Development Center in Advanced Telecommunications. He has been a Principal Investigator with the Signal Processing in Communications Group, University of Vigo, which participated in several European projects, including CERTIMARK, ECRYPT, REWIND, NIFTY, WITDOM, UNCOVER, and TRUMPET. He has coauthored over 70 papers in leading international journals, 180 peer-reviewed conference papers, and several international patents. His research interests include the areas of digital communications, adaptive algorithms, privacy enhancing technologies, and information forensics and security.

Dr. Pérez-González was an Associate Editor of IEEE SIGNAL PROCESSING LETTERS from 2005 to 2009 and IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY from 2006 to 2010, a Senior Area Editor of IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY from 2019 to 2022, and the Editor-in-Chief of the *EURASIP Journal on Information Security* from 2017 to 2022.