Improving PRNU Compression through Preprocessing, Quantization and Coding

Luca Bondi, *Student Member, IEEE,* Paolo Bestagini, *Member, IEEE,* Fernando Pérez-González, *Fellow, IEEE,* and Stefano Tubaro, *Senior Member, IEEE*

Abstract—In last decade the extremely rapid proliferation of digital devices capable of acquiring and sharing images over the Web has significantly increased the amount of digital images publicly accessible by everyone with Internet access. Despite the obvious benefits of such technological improvements, it is becoming mandatory to verify the origin and trustfulness of such shared pictures. Photo Response Non-Uniformity (PRNU) is the reference signal for forensic investigators when it comes to verifying or identifying which camera device shot a picture under analysis. In spite of this, PRNU is almost a white-shaped noise, thus being very difficult to compress for storage or large scale search purposes, which are frequent investigation scenarios. To overcome the issue, the forensic community has developed a series of compression algorithms. Lately, Gaussian Random Projections have proved to achieve state-of-the-art performance. In this paper we propose two additional steps that help improving even more Gaussian Random Projections compression rate: i) a decimation preprocessing step tailored at attenuating frequency components in which PRNU traces are already suppressed in JPEG compressed images; ii) a dead-zone quantizer (rather than the commonly used binary one) that enables an entropy coding scheme to save bitrate when storing PRNU fingerprints or sending residuals over a communication channel. Reported results show the effectiveness of proposed improvements, both under controlled JPEG compression and in a real case scenario.

Index Terms—PRNU preprocessing, PRNU interpolation, JPEG low pass, random projections, dead-zone quantization

I. INTRODUCTION

The amount of multimedia content generated and shared everyday through social media (e.g., Facebook, Instagram, Twitter, etc.) or photo sharing platforms (e.g., Flickr, Snapchat, Pinterest, etc.) is rapidly increasing, mainly thanks to easy access to audio-visual recording technologies and mobile Internet connections. Images and videos are acquired and shared on the web everyday by both professionals and amateurs, and their widespread diffusion has posed many challenges in terms of security. As a matter of fact, fast access to multimedia digital contents and ease of manipulation made development of digital forensics investigation methods an urgent necessity [1], [2]. When contents produced with imaging devices are used with malicious or criminal intent, it is paramount to perform forensic analyses to assess their authenticity and integrity, and eventually prevent their diffusion. In this scenario, a common problem faced by forensic investigators is camera source attribution, i.e., to identify which device took a given picture [3]. Courts, police departments, newspapers and companies are just some of the parties interested in solving ownership attribution disputes and verify the authenticity of images.

Active forensic methods based on image watermarking or bit-stream embedded metadata are not always applicable for owner identification purposes, or may be easy to fool. Watermarks, for instance, must be inserted at image inception time to let the content be recognizable even after common transformations such as rotation, scaling, or compression [4]. Conversely, metadata manipulation is at everyone's hand, and with a few clicks it is possible to anonymize an image removing all its EXIF properties [5].

To face ownership attribution in a completely blind fashion, researchers have developed a set of techniques tailored to extract traces left on the image by processing components such as lens aberrations [6], [7], color filter array (CFA) demosaicing artifacts [8], [9], JPEG compression traces [10], or combination of these [11], [12].

In addition to all the aforementioned blind methods, Photo Response Non-Uniformity (PRNU) gained a lot of interest in the last years [13], [14], [15], [16] due to its intrinsic robustness. PRNU traces are mainly due to sensor's silicon imperfections and can be exploited as a unique fingerprint for each camera sensor. Such a fingerprint is embedded in every shot taken with a specific device, being it a professional Digital Single-Lens Reflex (DSLR) camera or a cheap smartphone.

As a matter of fact, the PRNU fingerprint extraction process from natural images has been deeply analyzed [17], [18], [19], [20], [21], [22] and the effect of image content residuals after noise extraction has been faced with signal enhancement techniques [23]. Moreover, native robustness of extracted PRNU traces to cropping, scaling and JPEG compression [24], [25] gave way to a massive usage of PRNU fingerprints for camera device identification [26]. Furthermore, clustering

based approaches [27], [28] have shown the possibility of separating pictures coming from several camera devices based on the analysis of PRNU traces.

Unfortunately, one of the issues in performing device attribution through PRNU in real-world applications, is the nature of PRNU signal itself. Indeed, PRNU is characterized as a weak multiplicative noise-like signal, deterministic but hard to efficiently compress in a lossless fashion. In a large-scale retrieval setup the need of storing several thousands of reference fingerprints at full resolution poses issues regarding the amount of storage space. Moreover, when a noise-like residual extracted from a query image needs to be correlated with all the reference fingerprints in the database, a more compact version of both the fingerprint and the residual would reduce the computational cost. In a mobile authentication scenario [29], when a user wants to authenticate its device by sending a residual extracted from a picture to a centralized database for verification, restricting the amount of data being sent over the network is mandatory to reduce response times and improve user experience.

Several PRNU compression methodologies have been proposed in the literature. Fingerprint digest [30] is one of the most effective techniques, unfortunately bound to the knowledge of a strong reference PRNU fingerprint in order to determine positions of prominent peaks. A simple yet effective compression method is based on fingerprint binarization [31], which also provides benefits in computational terms by moving from a cross-correlation test to an Hamming distance computation. Several sub-linear hashing methods based on reference fingerprint digest [32], [33] have been developed to address the problem of fast search in large dataset. Again in [34], the problem of minimizing the number of observations required to reduce error probabilities below some pre-fixed misdetection rates is addressed in a Sequential Probability Ratio Test framework. As of now, state-of-the-art in PRNU compression is achieved by binarized Gaussian Random Projections by Valsesia et al. [35], [36]. This technique exploits compressive sensing concepts. The main idea is to preserve angles between PRNU fingerprint vectors when moving from a high dimensionality input space to a reduced dimensionality vector space, according to the Johnson-Lindenstrauss lemma. The effect of signal peaks selection after Random Projections has been studied in the form of binary adaptive embedding in [37], showing that a bias is introduced in the distance values.

In a recent paper [38] the authors faced the problem of defining a design principle for projection matrices tailored to PRNU compression. The baseline observation is that the point spread function of the autocorrelation function of PRNU fingerprints has a support that is always larger than one pixel. In particular, JPEG compression enlarges this support in a way that it is possible to design a very simple projection matrix that allows to get the same or better level of compression as Gaussian Random Projections at a computational cost below 0.1%.

In this paper we propose a methodology for PRNU compression tailored to JPEG images. This means reducing the size of PRNU databases, or residuals directly extracted from images under analysis, still ensuring high camera detection capability. This problem is faced considering two metrics: i) storage space/transmission bitrate; ii) computational complexity. These are essential to achieve high-compression rates while bounding camera attribution complexity.

In particular we leverage on two founding concepts:

- As images are often JPEG compressed, high-frequency PRNU components may be corrupted, thus making low-frequencies overall more informative.
- Dead-zone quantization provides much more flexibility than binary one, especially when paired with entropy coders.

Exploiting the first idea, we show that it is possible to pre-process PRNU traces with a decimation operation up to a certain ratio before reducing vector space dimensionality with Random Projections, still retaining important camera device information. Even though PRNU is robust to JPEG compression in terms of device identification or verification [13], the strong quantization introduced by JPEG compression at high frequency components [39], [40] motivates the introduction of a low-pass filtering via decimation as first step of the pipeline, in order to preserve only those frequency components that are carrying significant information about the original PRNU signal.

The second idea basically compromises between fingerprint binarization [31] (i.e., binary quantization) and fingerprint digest [30] (i.e., only coding prominent peaks). Thanks to the proposed dead-zone quantization process, it is possible to further compress fingerprints exploiting entropy coders that did not prove useful after PRNU binary quantization.

Experiments are carried out both under controlled JPEG compression, starting from RAW images from the Dresden Image Database [41], and in a wild scenario, with JPEG images compressed directly by camera's firmware. In all cases we take care of both detection accuracy and computational complexity of proposed improvements. Results show that, by using the proposed pre-processing and quantization scheme, it is possible to further push the performance of Gaussian Random Projections when JPEG images are actually considered, while achieving the state of the art when RAW images are available.

The rest of this work is organized as follows. Section II reports the necessary background concepts. Section III is dedicated to the formal problem definition and presentation of PRNU compression state-of-the-art techniques. Section IV explains the rationale behind the proposed pipeline and its algorithmic details. Section V describes the experimental setup for results reported in Section VI. Conclusions are drawn in Section VII.

II. BACKGROUND

In the following, we provide an overview of PRNU extraction techniques and the JPEG compression pipeline, which are useful to understand the rationale behind the proposed compression method.

A. Notation

Vectors are given by boldface letters, e.g., $\mathbf{x}$ and are considered to be column vectors. The ith sample of $\mathbf{x}$ is represented

by $x(i)$. Matrices are denoted by bold capital letters, e.g., $\mathbf{X}$, and the i,jth element is indicated by $X(i,j)$. Given a matrix $\mathbf{X}$, its column-wise unwrapped vector is denoted by $\mathbf{x}$. The Hadamard (sample-wise) product between $\mathbf{x}$ and $\mathbf{y}$ is denoted by $\mathbf{x} \circ \mathbf{y}$. The sample-wise division between $\mathbf{X}$ and $\mathbf{Y}$ is denoted as $\mathbf{X} \div \mathbf{Y}$. The matrix multiplication between $\mathbf{X}$ and $\mathbf{Y}$ is denoted by $\mathbf{X} \odot \mathbf{Y}$.

B. PRNU extraction

Photo Response Non-Uniformity (PRNU) is a multiplicative noise pattern mostly related to different sizes of imaging sensor cells. It is a weak signal caused by minute imperfections occurring during the manufacturing process of the sensor. Despite its weakness, when a sufficiently large number of image samples is available it is possible to estimate and use it as a robust fingerprint for a specific camera sensor [17].

PRNU extraction is based on a simplified linear model of camera sensor output [15]

$$\mathbf{I} = g^\gamma \left[(1 + \mathbf{K}_0)\mathbf{Y} + \mathbf{\Lambda} \right]^\gamma + \mathbf{\Theta}_q \qquad (1)$$

where $\mathbf{I}$ is an $h \times w$ matrix of the same size in pixels of the sensor, g is the color channel gain and γ is the gamma correction factor. $\mathbf{K}_0$ is a zero-mean noise-like signal responsible for the PRNU fingerprint. $\mathbf{\Lambda}$ is a combination of remaining noise sources (dark currents, read-out noise, shot noise) and $\mathbf{\Theta}_q$ is the quantization noise. As shown in [15], the imaging model can be further simplified as

$$\mathbf{I} = \mathbf{I}^{(0)} + \mathbf{I}^{(0)}\mathbf{K} + \mathbf{\Theta} \qquad (2)$$

where $\mathbf{I}^{(0)} = (g\mathbf{Y})^\gamma$ is the noiseless image, $\mathbf{K} = \gamma\mathbf{K}_0$ and $\mathbf{\Theta} = \gamma\mathbf{I}^{(0)}\mathbf{\Lambda}/\mathbf{Y} + \mathbf{\Theta}_q$ condensates independent random noise components.

The first step toward PRNU estimation is a noise extraction process that aims at preserving only noise-like residuals from $\mathbf{I}$, thanks to a properly designed filter. One of the most used noise extraction algorithms [14] is based on multi-level wavelet noise-enhancement via adaptive Wiener filtering and it is applied to red, green and blue channels of $\mathbf{I}$ separately to obtain a residual $\mathbf{W}$.

When a set of F_c shots $\mathbf{I}_k$ with $k = 1, \ldots, F_c$ taken from the same camera c is available, the Maximum Likelihood estimate for $\mathbf{K}_c$ results in

$$\hat{\mathbf{K}}_c = \left(\sum_{k=1}^{F_c} \mathbf{W}_k \circ \mathbf{I}_k \right) \div \left(\sum_{k=1}^{F_c} \mathbf{I}_k^2 \right) \qquad (3)$$

When only a single query image $\mathbf{I}^{(q)}$ is available we call its residual $\mathbf{W}^{(q)}$.

A few post-processing operations on $\hat{\mathbf{K}}_c$ and $\mathbf{W}^{(q)}$ are then applied in order to remove the average from each color channel, subtract row and column means for each color component separately and finally merge the three color components. At last a noise-peaks removal step is applied as a Wiener filtering in the discrete Fourier domain, to remove residual periodicities artifacts and whiten the spectrum of the resulting fingerprint or residual estimate.

Given a camera device c characterized by a PRNU fingerprint $\mathbf{K}_c$, and a query image $\mathbf{I}^{(q)}$, whose residual is $\mathbf{W}^{(q)}$, a binary hypothesis testing problem defined as

$$H_0 : \mathbf{I}^{(q)} \text{ was not taken with camera } c$$
$$\text{thus it does not contain } \mathbf{K}_c$$
$$H_1 : \mathbf{I}^{(q)} \text{ was taken with camera } c$$
$$\text{thus it contains } \mathbf{K}_c$$

is faced in order to determine whether the query image has been shot with the given camera device. Detection of such matching can be performed via a cross-correlation test, defined as

$$\rho\left(\hat{\mathbf{K}}_c, \mathbf{W}^{(q)} \right) = \sum_{i=1}^{h} \sum_{j=1}^{w} \hat{K}_c(i,j) \cdot W^{(q)}(i,j) \qquad (4)$$

When $\rho\left(\hat{\mathbf{K}}_c, \mathbf{W}^{(q)} \right) > \tau$ then $\mathbf{I}^{(q)}$ is decided to contain $\mathbf{K}_c$, thus the query image is attributed to camera c (verifying H_1 hypothesis). τ is a threshold properly set in order to bound the false-alarm probability under a desired target value.

C. JPEG compression

JPEG compression is the most widespread standard for saving natural pictures in a digitized way. All camera models and smartphones, both professional and cheap ones, provide a way to save on non-volatile storage the acquired images in JPEG format.

At first, a color space transformation from the RGB color space to the YCbCr color space is applied to the original image, to get a luma component (Y) and two chroma components (Cb, Cr). Luma and chroma matrices are split in 8×8-pixel non-overlapping blocks. Every block is transformed with 2D Discrete Cosine Transform, rearranging the 64 resulting coefficients in a 8×8 matrix where the top-left element contains the DC component and the bottom-right element contains the highest – both vertically and horizontally – frequency component coefficient.

Each block is then quantized by dividing each frequency component by a specific quantization step, then rounding the result to the nearest integer. Quantization coefficients are stored in two quantization matrices, one for luma component and one for chroma components. The aforementioned coefficients quantization carries two effects: i) a reduction of inter-coefficient entropy, exploited by zig-zag run-length Huffman coding to compress each block thus reducing storage space; ii) a frequency-dependent filtering, that greatly reduces high-frequency components while preserving low-frequency ones.

The final effect of JPEG compression on an image is a block-wise low-pass filtering. This also affects the image embedded PRNU, which loses its white-shaped spectrum in favor of a low-pass version. This consideration stands behind the choice of pre-processing with a low-pass filter the estimated PRNU fingerprints and image residuals, in order to reduce the amount of data being processed and transmitted, while preserving the surviving spectral components of the PRNU.
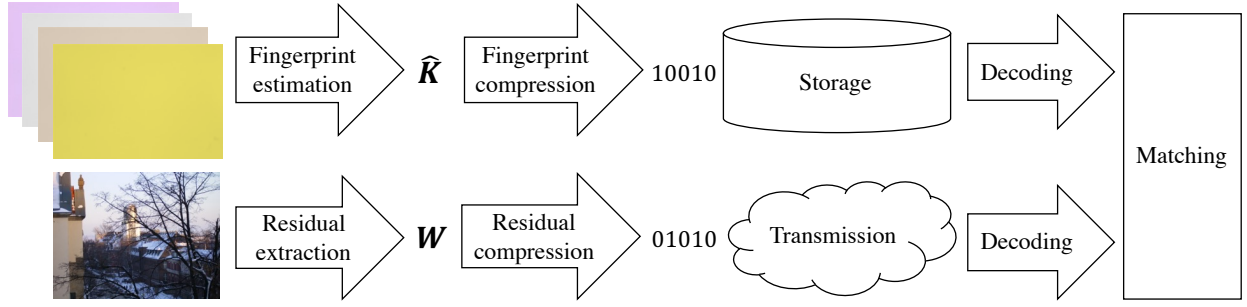
Fig. 1: Overall database and query pipeline. (Top) A set of flatfield images is used to estimate a camera fingerprint $\hat{K}$, that is compressed and stored into a database. (Bottom) Residual $\mathbf{W}^{(q)}$ is extracted from a single query image, then compressed and sent to a central location for matching purpose.

III. PRNU COMPRESSION

In this section, we introduce the PRNU fingerprint and residual compression problem, depicting two commonly considered scenarios. An overview of lossy compression methods developed over the last years follows.

A. Problem Formulation

When it comes to storing a huge amount of fingerprints or there is the need of sending them over a band-limited communication channel, an effective PRNU compression method becomes mandatory.

We are interested in two main applicative scenarios that can summarize several real-world applications (see Figure 1). Both scenarios include two players: i) a central database that stores camera fingerprints, each extracted from several images; ii) a number of query devices whose fingerprints need to be sent to a central server for matching purposes. The two scenarios are described in the following:

Query compression scenario: the goal is to reduce as much as possible the amount of memory used to represent query residual information. This means minimizing the bitrate required to send the compressed residual from a remote device to a central server. Equivalently, this can be interpreted as minimizing the file size in case of residual storage applications.

Joint database-query compression scenario: the first goal is to restrict as much as possible both the bitrate required to send a query residual to a central server and at the same time limit the storage space required to store camera fingerprints. The second goal is to reduce the computational complexity required to match each camera fingerprint with a given query residual.

One of the main issues in compressing a PRNU fingerprint comes from the observation that it is an inherently broadband white noise-like signal, well modeled as a sequence of i.i.d. samples drawn from a zero mean Gaussian distribution. This results in a signal with little or no redundancies to be exploited for lossless compression. Lossy compression is then the only way to reduce fingerprint's rate. In particular, we focus on compressing PRNU fingerprints and residuals when no geometrical transformations are applied to the original images.

From a formal point of view, the problem of PRNU compression faced in this work can be defined as follows. Let $\mathcal{C}$ be a collection of camera fingerprints, where each fingerprint is estimated from several flatfield images according to (3). Let $\mathcal{Q}$ be a collection of query residuals, where each residual is extracted from a single query image. Let $\hat{\mathbf{K}}_c$ and $\mathbf{W}^{(q)}$ be an estimated fingerprint and a noise residual extracted respectively from $\mathcal{C}$ and $\mathcal{Q}$. The main goal is to generate reduced rate representations of $\hat{\mathbf{K}}_c$ and $\mathbf{W}^{(q)}$ such that

- the performances in terms of Receiver-Operating-Characteristic of the compressed and the uncompressed case are similar.
- storage space – or transmission rate – requirements of the compressed fingerprints and residuals are minimized.

B. State-of-the-art Compression Methods

Due to the ever increasing demand of higher compression rates, several techniques have been proposed in the literature to work either in the PRNU fingerprint or residual compression scenarios. The illustrated compression strategies are described in terms of camera fingerprint ($\hat{\mathbf{K}}_c$) but the same process holds also for query residuals ($\mathbf{W}^{(q)}$).

Trimming and cropping. Fingerprint trimming [30] is the most trivial way of compression. Considering $\hat{\mathbf{k}}_c$ as the column-wise unwrapping of $\hat{\mathbf{K}}_c$, trimming is performed by preserving only the first P samples from $\hat{\mathbf{k}}_c$. Similarly, fingerprint cropping results when preserving only the central portion of a fingerprint $\hat{\mathbf{K}}_c$ and then performing the unwrapping. This is the reference baseline method.

Digest. Fingerprint digesting [30] comes from the idea that most prominent peaks of the extracted PRNU $\hat{\mathbf{K}}_c$ are more relevant when using (4) for camera attribution. Therefore, the digest is built by retaining the position and value of the P highest energy pixels from $\hat{\mathbf{K}}_c$, creating a pair of vectors of length P, holding respectively peak values and positions. While this method turns out to be very effective in terms of compression ratio, it requires the knowledge of a rather good estimate of $\hat{\mathbf{K}}_c$ to preserve those pixels that are really characterizing a specific sensor fingerprint. This is made possible only when $\hat{\mathbf{K}}_c$ is estimated from many images, as only with a good estimate of the PRNU the peaks selection process is robust enough to allow compression while retaining high detection performance. In a query compression scenario, as in a joint compression scenario, where query residual

compression is limited to the knowledge of a single image, fingerprint digesting is not a viable option.

Gaussian Random Projections. Introduced as PRNU compression method by Valsesia et al. [36], Random Projections (RP) with Gaussian sensing matrix have proven to be an effective way of compressing camera fingerprints and query residuals. The idea of RP is to project the one dimensional unwrapping $\hat{\mathbf{k}}_c$ of fingerprint $\hat{\mathbf{K}}_c$ from a vector space of dimension $L = h \cdot w$ to a subspace of dimension P.

Formally, a sensing (projection) matrix $\boldsymbol{\Phi}$ with size $L \times P$ is generated with samples being extracted from a i.i.d. zero-mean Gaussian distribution. The resulting projection $\mathbf{r}_c$ is thus a matrix product between a sensing matrix $\boldsymbol{\Phi}$ and a vector $\hat{\mathbf{k}}_c$

$$\mathbf{r}_c = \boldsymbol{\Phi} \odot \hat{\mathbf{k}}_c \tag{5}$$

In order to speed-up computation and save memory, a simplified way of building the projection matrix is to randomly generate as i.i.d. zero-mean Gaussian a single column ϕ of $\boldsymbol{\Phi}$, then generate all other columns by means of circular shifting. In this way the projection can be turned into an element-wise product in the Fourier Transform domain as

$$\mathbf{r}_c = \text{IFFT}\left(\text{FFT}(\hat{\mathbf{k}}_c) \circ \text{FFT}(\phi)\right) \tag{6}$$

The process ends by preserving only the first P elements of $\mathbf{r}_c$. The binary version of $\mathbf{r}_c$ is denoted as r_c^b.

Binarization Fingerprint binarization [31] is an effective way to greatly reduce the fingerprint bitrate even after trimming, cropping or projection. Binarization is defined as an element-wise operation transforming a real number x into its binarized version x_b as

$$x_b = \begin{cases} +1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases} \tag{7}$$

An additional benefit from binarization is the reduced computational complexity when performing a cross-correlation test, as shown by Bayram et al. [31].

IV. PROPOSED COMPRESSION PIPELINE

Given a camera fingerprint $\hat{\mathbf{K}}_c$ acquired according to Eq. (3) and a query residual $\mathbf{W}^{(q)}$ we propose the same compression pipeline for both $\hat{\mathbf{K}}_c$ and $\mathbf{W}^{(q)}$. For the sake of clarity, in the following we describe the process only for $\hat{\mathbf{K}}_c$, using $\hat{\mathbf{K}}$ as a short notation for $\hat{\mathbf{K}}_c$.

Algorithm 1 and Figure 2 depict the proposed approach, comprising four steps: i) fingerprint $\hat{\mathbf{K}}$ is decimated over rows and columns by a factor d to generate $\hat{\mathbf{K}}_d$; ii) Random Projections (RP) are applied to $\hat{\mathbf{K}}_d$ to produce a vector $\mathbf{r}_P^*$ of length P; iii) dead-zone (DZ) quantization with threshold δ preserves only the peaks of $\mathbf{r}_P^*$ and creates $\mathbf{r}_P^\delta$; iv) Entropy Coding (EC) applied to $\mathbf{r}_P^\delta$ generates the compressed output bit-stream containing fingerprint information. In the following we illustrate the rationale and the details for each step, introducing compression for the case of the fingerprint estimate $\hat{\mathbf{K}}$. The whole pipeline holds exactly in the same way also for query residuals $\mathbf{W}^{(q)}$.

Algorithm 1 Camera fingerprint processing algorithm

Require: $\hat{\mathbf{K}}$, d, ϕ, P, δ
  $\hat{\mathbf{K}}_{\text{tmp}} \leftarrow$ DECIMATE($\hat{\mathbf{K}}$, d)
  $\check{\mathbf{K}} \leftarrow$ DECIMATE($\hat{\mathbf{K}}_{\text{tmp}}$, d)
  $\check{\mathbf{k}} \leftarrow$ UNWRAP($\check{\mathbf{K}}$)
  $\mathbf{r}^* \leftarrow$ RANDOM PROJECT($\check{\mathbf{k}}$, ϕ, P)
  $\mathbf{r}^\delta \leftarrow$ DEAD-ZONE QUANTIZE($\mathbf{r}$, δ)
  bit-stream $\leftarrow$ ENTROPY ENCODE($\mathbf{r}^\delta$)

function DECIMATE($\mathbf{A}$, d)
    $h, w \leftarrow$ SIZE($\mathbf{A}$)
    for i_r in $\{0, 1, \ldots, h - 1\}$ **do**
        for i_c in $\{0, 1, \ldots, \lfloor (w - 1)/d \rfloor\}$ **do**
            $A_d(i_r, i_c) = 0$
            for j_c in $\{0, 1, \ldots, w - 1\}$ **do**
                $x = i_c \cdot d - j_c$
                $A_d(i_r, i_c) = A_d(i_r, i_c) + h_c(x) \cdot A(i_r, j_c)$
            end for
        end for
    end for
    return $\mathbf{A}_d$
end function
function RANDOM PROJECT($\mathbf{A}$, ϕ, P)
    $\mathbf{a} \leftarrow$ FLATTEN($\mathbf{A}$)
    $\mathbf{a}^{\mathcal{F}} \leftarrow$ FFT($\mathbf{a}_d$), $\phi^{\mathcal{F}} \leftarrow$ FFT(Φ)
    $\mathbf{p}^{\mathcal{F}} \leftarrow \mathbf{a}^{\mathcal{F}} \circ \phi^{\mathcal{F}}$
    $\mathbf{p} \leftarrow$ IFFT($\mathbf{p}^{\mathcal{F}}$)
    $\mathbf{p} \leftarrow$ TRIM($\mathbf{p}$, P)
    return $\mathbf{p}$
end function
function DEAD-ZONE QUANTIZE($\mathbf{a}$, δ)
    $P \leftarrow$ LENGTH($\mathbf{a}$), $\sigma \leftarrow$ STD($\mathbf{a}$)
    for i in $\{0, \ldots, P - 1\}$ **do**
        if $a(i) > \delta\sigma$ **then**
            $a_q(i) = +1$
        else if $a(i) < -\delta\sigma$ **then**
            $a_q(i) = -1$
        else
            $a_q(i) = 0$
        end if
    end for
    return $\mathbf{a}_q$
end function
function ENTROPY ENCODE($\mathbf{a}$)
    $P \leftarrow$ LENGTH($\mathbf{a}$)
    for i in $\{0, \ldots, P - 1\}$ **do**
        if $a(i) = 1$ **then**
            bitstream $\leftarrow$ bitstream $+10$
        else if $a(i) = -1$ **then**
            bitstream $\leftarrow$ bitstream $+11$
        else
            bitstream $\leftarrow$ bitstream $+0$
        end if
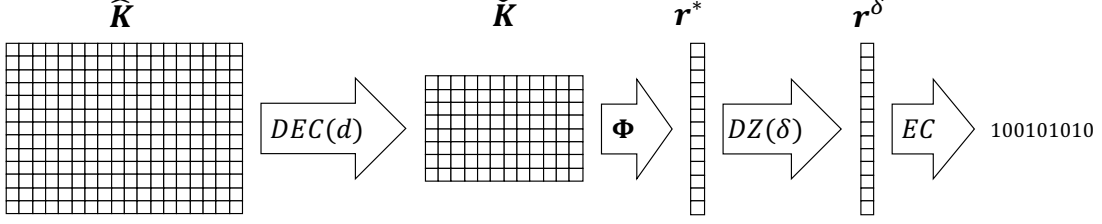   **end for**
    return bitstream
end function

Fig. 2: Proposed compression pipeline. A camera fingerprint estimate $\hat{\mathbf{K}}$ is first decimated (DEC) by a factor d to obtain $\check{\mathbf{K}}$, then Random Projections (RP) are used to compress $\check{\mathbf{K}}$ into a P elements vector $\mathbf{r}^*$. Finally a dead-zone quantizer (DZ) is applied with threshold δ to get a quantized fingerprint $\mathbf{r}^\delta$ successively encoded (EC) to generate a compressed output bit-stream.
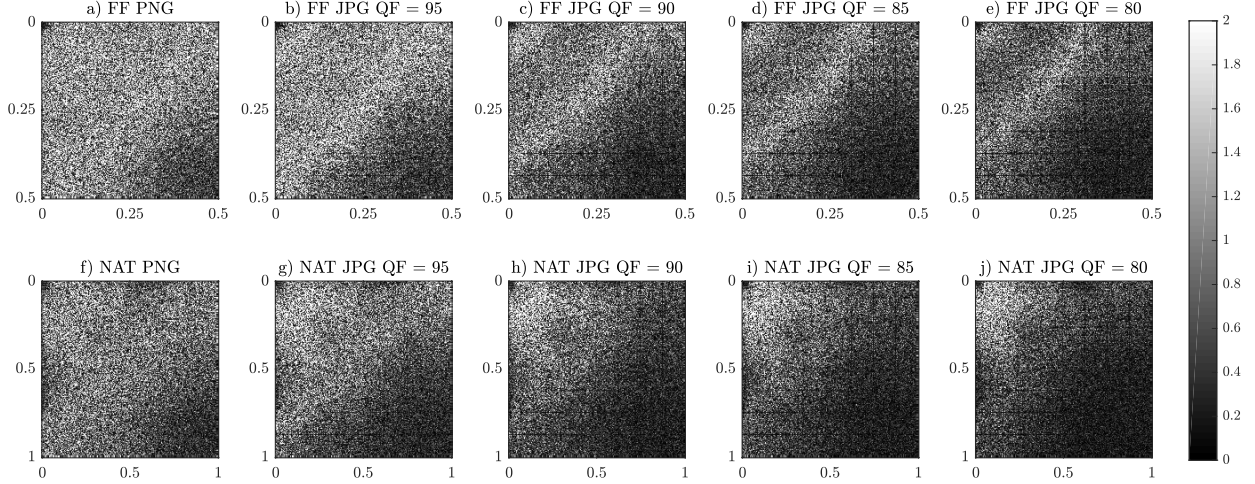

Fig. 3: Power Spectral Density (PSD) for noise residuals from a single flatfield image (a,b,c,d,e) and from a single natural image (f,g,h,i,j) on PNG uncompressed images and while varying JPEG quality factor.

A. Decimation

It is known from [40] that JPEG compression increases the variance of cross-correlation values in (4), thus reducing the margin between H_0 and H_1 hypotheses. In Figure 3 we analyze the effect of JPEG compression on the Power Spectral Density (PSD) for different quality factors by looking at noise residuals $\mathbf{W}$ extracted from flatfield and natural images. It is clear that the power of the residue at high spatial frequencies – lower right quadrant – lowers as images are more compressed. Moreover, residual PRNU contributions in high-frequency bins are combined with residuals of blockiness artifacts from JPEG compression that cannot be completely removed by the residue extraction process.

Putting together the aforementioned consideration, a reasonable and simple preprocessing method to reduce the dimensionality of $\hat{\mathbf{K}}$ and attenuate its high-frequency components consists in decimating $\hat{\mathbf{K}}$ by a factor $d > 1$ along rows and columns. This operation is accomplished via interpolation with a cubic kernel [42] $h_c(x)$ defined as

$$h_c(x) = \begin{cases} 1.5|x|^3 - 2.5|x|^2 + 1 & \text{if } |x| \leq 1 \\ -0.5|x|^3 + 2.5|x|^2 - 4|x| + 2 & \text{if } 1 < |x| \leq 2 \\ 0 & \text{otherwise} \end{cases}$$

(8)

Given a vector $\mathbf{a}$ of length L and a decimation factor d, the i-th element of its decimation a_d results in

$$a_d(i) = \sum_{j=0}^{L-1} h_c(j - i \cdot d) \cdot a(j), \forall i \in \{0, \ldots, \lfloor l/d \rfloor\}$$

(9)

The choice of d is carried out such that the resulting resized fingerprint $\check{\mathbf{K}}$ performs in the same way as the original $\hat{\mathbf{K}}$ fingerprint in terms of detection performance.

B. Random Projection

The second step in the pipeline consists in projecting $\check{\mathbf{k}}$ – the column-wise unwrapping of $\check{\mathbf{K}}$ – with P Random Projections according to (6), to obtain $\mathbf{r}^*$. Recalling that $L = w \cdot h$ is the number of pixels in the sensor, it is worth observing that the input to Random Projection is now a vector with L/d^2 elements due to the previous decimation, thus the computational complexity in terms of additions and multiplications of implementing (6) for $\check{\mathbf{k}}$ is

$$\mathrm{C}_{RP}(\check{\mathbf{k}}) = 2\frac{L}{d^2}\left[\log_2(L) + 3 - 2\log_2(d)\right]$$

(10)

whereas the computational complexity of implementing (6) for $\hat{\mathbf{k}}$ – the column-wise unwrapping of $\hat{\mathbf{K}}$ – as in [36] is

$$\mathrm{C}_{RP}(\hat{\mathbf{k}}) = 2L\left[\log_2(L) + 3\right]$$

(11)

Finally observing that

$$\mathrm{C}_{RP}(\check{\mathbf{k}}) < \frac{1}{d^2}\mathrm{C}_{RP}(\hat{\mathbf{k}}) \qquad (12)$$

we can conclude that the computational complexity is reduced by more than a factor d^2.

C. Dead-zone quantization

While binarization of random projections has been proved to be an effective way of quantizing and preserving good performance in terms of detection, here we propose to use a dead-zone quantizer on $\mathbf{r}^*$ to get $\mathbf{r}^\delta$. Given σ, the standard deviation of $\mathbf{r}^*$, the i-th element of $\mathbf{r}^\delta$ for $i = 1, \ldots, P$ is obtained as

$$r^\delta(i) = \begin{cases} +1 & \text{if } r^*(i) > \delta\sigma \\ 0 & \text{if } -\delta\sigma \le r^*(i) \le \delta\sigma \\ -1 & \text{if } r^*(i) < -\delta\sigma \end{cases} \qquad (13)$$

The rationale behind this choice is twofold. On one hand, as observed in [30] for PRNU digest, peaks with high absolute values are the most important ones in terms of cross-correlation, thus preserving those peaks seems a reasonable choice. On the other hand, quantizing with a variable threshold allows to reduce the bitrate of $\mathbf{r}^\delta$ via entropy coding by fixing P while increasing the value of δ (i.e., setting more samples to zero). When comparing a dead-zone quantized signal with another binarized or dead-zone quantized signal, the similarity measure provided by the cross-correlation, Eq. (4) is equivalent to the Opposite Absolute Distance (OAD) defined as:

$$\mathrm{OAD}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} 1 - |\mathbf{x}(i) - \mathbf{y}(i)| \qquad (14)$$

where $\mathbf{x}$ and $\mathbf{y}$ are respectively the two binarized or dead-zone quantized reference fingerprint and query residuals of length N.

D. Entropy coding

As last step of the pipeline, an arithmetic entropy coding scheme is applied to $\mathbf{r}_P^\delta$ in order to get a compressed bit-stream containing compressed information about $\hat{\mathbf{K}}$. The threshold value δ is a tunable parameter, as will be discussed in Section VI. The Gaussian distribution of PRNU fingerprint coefficients after projection comes at help in the entropy coding stage. In fact, dead-zone quantization produces a three-symbols output whose entropy is lower than $\log_2(3)$ for increasing values of δ, due to the higher probability of symbol 0 with respect to symbols $+1$ and -1.

V. EXPERIMENTAL SETUP

In the following we provide details about the evaluation metrics and datasets adopted to provide experimental results in Section VI. To validate the effectiveness of the proposed method, we focus on the problem of image source attribution in a probabilistic framework with the following constraints:

- All extracted fingerprints $\hat{\mathbf{K}}_c$ and residuals $\mathbf{W}^{(q)}$ are cropped to their central region of size $h = w = 1500$,

thus $L = 2.25 \cdot 10^6$. This allows for a direct comparison between every fingerprint-residual pair.
- We consider only aligned fingerprints and residuals at original resolution, meaning that we are not looking for rotation, cropping, or other affine transformations. All fingerprints and residuals have the same size and for each camera device the cropped region offset with respect to the origin is fixed. This choice follows from state of the art works about PRNU compression [31], [36].

Given a set of C camera devices, for each device $c \in [1, C]$ we have F_c flatfield pictures that we use to estimate the camera fingerprint $\hat{\mathbf{K}}_c$, according to (3). In this way we build a dataset $\mathcal{C}$ of N_c device fingerprints. For each natural image $\mathbf{I}^{(n)}$, $n \in [1, N]$ we estimate its residual $\mathbf{W}^{(n)}$, building a dataset $\mathcal{Q}$ of N query images.

A. Evaluation metrics

In order to determine whether a query image residual $\mathbf{W}^{(n)}$ from $\mathcal{Q}$ is correctly binded to its camera device and not to other devices, we build a cross-correlation matrix $\mathbf{NCC} \in \mathbb{R}^{C \times N}$ defined as

$$CC(c,n) = \left\langle \hat{\mathbf{K}}_c \circ \mathbf{I}^{(n)}, \mathbf{W}^{(n)} \right\rangle \qquad (15)$$
$$c = 1, \ldots, C, \ n = 1, \ldots, N$$

then we set the same cross-correlation threshold τ_c for all camera devices such that the overall False-Positive rate is below a certain false-alarm probability p_{FA}. The cross-correlation matrix $\mathbf{CC}$ is then turned into a binary prediction matrix $\mathbf{P} \in \{0,1\}^{C \times N}$ according to

$$P(c,n) = \begin{cases} 1 & CC(c,n) > \tau_c \\ 0 & CC(c,n) \le \tau_c \end{cases} \qquad (16)$$
$$c = 1, \ldots, C, \ n = 1, \ldots, N$$

Comparison between $\mathbf{P}$ and the Ground-Truth binary matrix $\mathbf{GT}$, where $GT(c,n) = 1$ when $\hat{\mathbf{K}}_c$ and $\mathbf{W}^{(n)}$ are from the same camera device, leads to definition of the hereinafter used evaluation metrics known as True-Positive Rate (TPR) and False-Positive Rate (FPR). In particular, when evaluating the relationship between residual bitrate and system performance, we are considering the True-Positive Rate at a specific false-alarm probability $p_{\mathrm{FA}} = 0.05$.

B. Datasets

Resorting to images from the Dresden Image Database [41] we build two camera fingerprint datasets and several query residual datasets.

Four controlled compression datasets are built upon RAW images coming from 6 camera devices, two for each model of *Nikon-D200*, *Nikon-D70*, *Nikon-D70s*:
- $\mathcal{C}_f^{\mathrm{RAW}}$ is composed of 6 camera fingerprints extracted from flatfield RAW images.
- $\mathcal{Q}_n^{\mathrm{RAW}}$ is composed of 1317 query residual extracted from natural RAW images.
- $\mathcal{Q}_n^{\mathrm{QF}=q}$ is composed of 1317 query residual extracted from natural RAW images compressed in JPEG format with QF $= q$ before the noise extraction process.
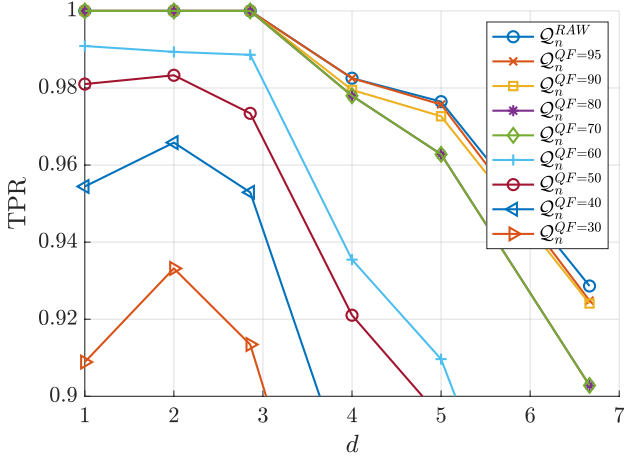
Fig. 4: Effect of decimating by factor d in terms of detection performance when query images are uncompressed ($\mathcal{Q}_n^{\mathrm{RAW}}$) or JPEG compressed with different quality factors ($\mathcal{Q}_n^{\mathrm{QF=q}}$, $q \in \{30, 40, 50, 60, 70, 80, 95\}$).
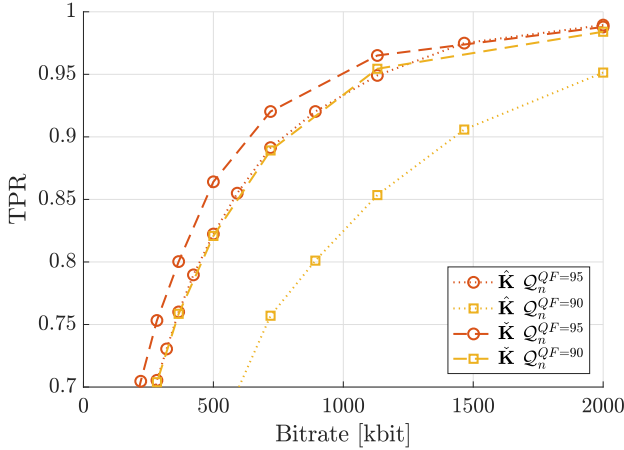

Fig. 5: Comparison between decimation with a factor $d = 2$ followed by central cropping ($\check{\mathbf{K}}$) against central cropping alone ($\hat{\mathbf{K}}$) in terms of detection performance when query images are JPEG compressed with different quality factors ($\mathcal{Q}_n^{\mathrm{QF=95}}$, $\mathcal{Q}_n^{\mathrm{QF=90}}$).
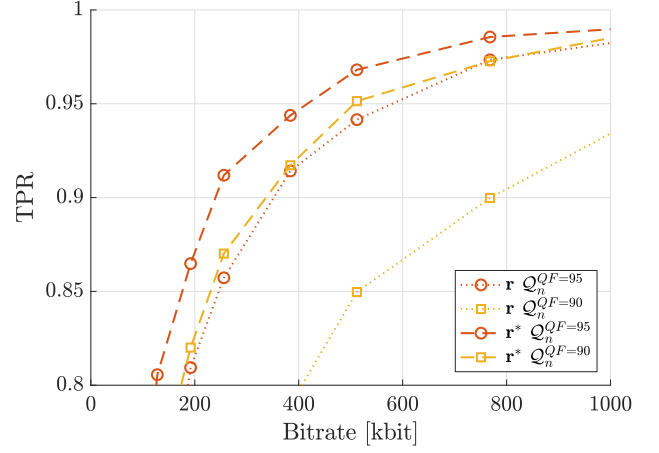

Fig. 6: Effect of decimation with a factor $d = 2$ followed by Random Projections in terms of detection performance when query images are JPEG compressed with different quality factors ($\mathcal{Q}_n^{\mathrm{QF=95}}$, $\mathcal{Q}_n^{\mathrm{QF=90}}$).
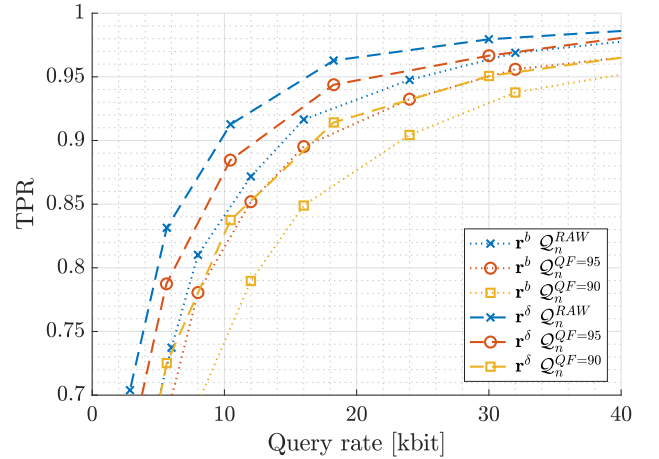

Fig. 7: Effect of dead-zone quantization, applied after decimation and Random Projection, in terms of detection performance when query images are uncompressed ($\mathcal{Q}_n^{\mathrm{RAW}}$) or JPEG compressed with different quality factors ($\mathcal{Q}_n^{\mathrm{QF=95}}$, $\mathcal{Q}_n^{\mathrm{QF=90}}$).

Two uncontrolled compression datasets are built upon JPEG images from 53 camera models, the same used in [36]:

- $\mathcal{C}_f^{\mathrm{JPG}}$ is a composed of 53 camera fingerprints extracted from flatfield JPEG images as encoded by cameras' firmware.
- $\mathcal{Q}_n^{\mathrm{JPG}}$ is a composed of 9092 query residual extracted from natural JPEG images as encoded by cameras' firmware.

VI. RESULTS

In the following we report experimental results. At first we show how to properly select the decimation kernel and factor. Then we compare Random Projections with and without the proposed resizing and dead-zone quantization approaches in terms of bitrate vs. True-Positive Rate. Finally we show how the proposed pipeline compares with the state of the art solution in terms of Receiver-Operating-Characteristic.

A. Decimation

Choice of d, the resizing factor for the first step of the pipeline, is performed evaluating the impact in terms of TPR when database fingerprints are extracted from $\mathcal{C}_f^{\mathrm{RAW}}$, while query residuals are extracted from $\mathcal{Q}_n^{\mathrm{RAW}}$ and $\mathcal{Q}_n^{\mathrm{QF=q}}, q \in \{30, 40, 50, 60, 70, 80, 95\}$. Figure 4 depicts the TPR at fixed $p_{\mathrm{FA}} = 0.05$, as a function of d. For weak JPEG compression (QF ≥ 70) we observe a drop in detection performance when decimating with a factor $d \geq 3$, while for $d < 3$ the accuracy is preserved almost without loss at a value of 1.0. For stronger JPEG compression factors (QF < 70) decimation with $d = 2$ results beneficial, as it increases the Signal-to-Noise Ratio between the PRNU (signal) and the PRNU-unrelated noise components remaining after the noise extraction process. It is also interesting to notice that the loss-less behavior of decimation with $d = 2$ might be related to CFA interpolation, even though we have no experimental evidences to prove it at this time. Given the aforementioned considerations we chose to set $d = 2$ for all the following experiments. As shown in
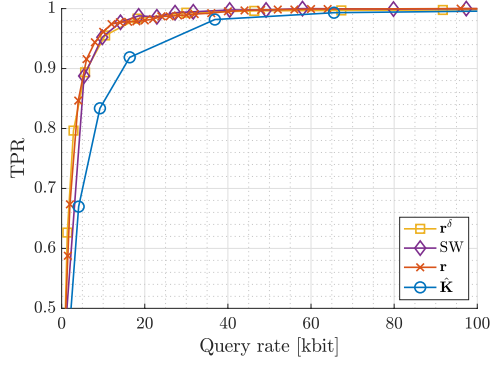
Fig. 8: Detection performance in a query compression scenario on uncompressed query images.
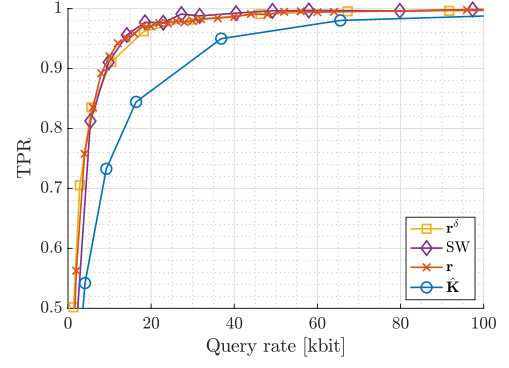

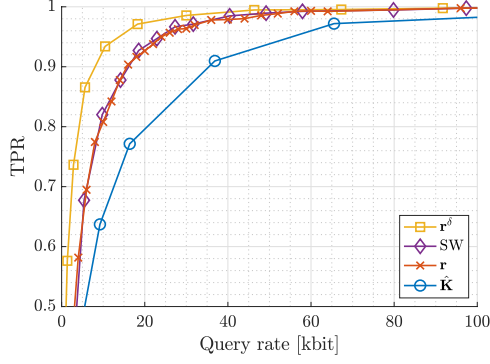Fig. 12: Detection performance in a joint compression scenario on uncompressed query images.

Fig. 9: Detection performance in a query compression scenario on JPEG compressed query images (QF = 95).
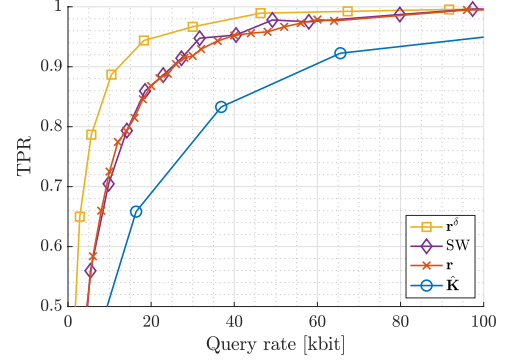

Fig. 13: Detection performance in a joint compression scenario on JPEG compressed query images (QF = 95).
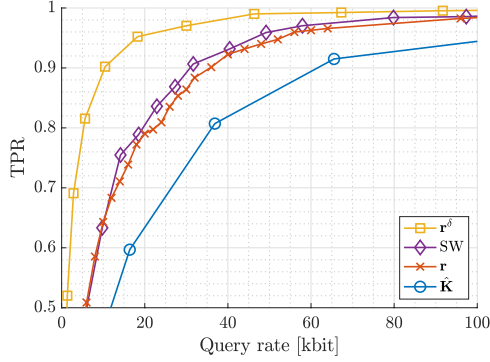

Fig. 10: Detection performance in a query compression scenario on JPEG compressed query images (QF = 90).
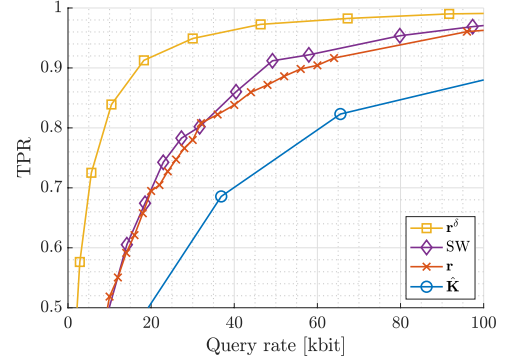

Fig. 14: Detection performance in a joint compression scenario on JPEG compressed query images (QF = 90).
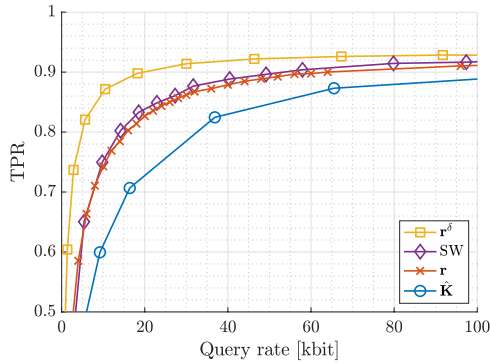

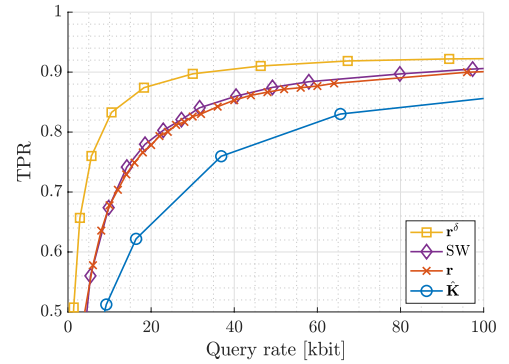Fig. 11: Detection performance in a query compression scenario on Dresden dataset.

Fig. 15: Detection performance in a joint compression scenario on Dresden dataset.

Sec. IV this leads to a 75% complexity reduction in terms of subsequent Random Projections. As for the choice of the kernel function, the cubic one defined in Eq. (8) shows similar detection rates when compared to Lanczos kernels, with a noticeable improvement with respect to a bilinear kernel, as it should be expected.

To understand the effect introduced by decimation when dealing with JPEG compressed query images, Figure 5 reports the comparison between a baseline central cropping strategy ($\hat{\mathbf{K}}$ dotted lines) versus a compression approach based on decimation by a fixed factor $d = 2$ followed by central cropping ($\check{\mathbf{K}}$ dashed lines). To vary the bitrate when no decimation is applied ($\hat{\mathbf{K}}$ dotted lines) we centrally crop both the fingerprint and the residual. When decimation of a fixed factor $d = 2$ is applied as a pre-processing step ($\check{\mathbf{K}}$ dashed lines) the bitrate is varied by central cropping both the decimated fingerprint and the decimated residual. The bitrate is computed as $l^2 \cdot 32$bit, where l is the side-length of the cropping square. Database images are drawn from $\mathcal{C}_f^{\mathrm{RAW}}$ while query images are extracted from $\mathcal{Q}_n^{\mathrm{QF=95}}$ and $\mathcal{Q}_n^{\mathrm{QF=90}}$. Results show that when JPEG query images are involved the same TPR can be obtained with a significantly lower bitrate, meaning that the interpolation effect introduced by the cubic kernel is preserving PRNU components and compacting them into a smaller support.

When Random Projections are used instead of central cropping, the benefits of decimation are confirmed and highlighted. Figure 6 shows the benefit of decimation with $d = 2$ when Random Projections are used to compress the signal while varying the projection space dimensionality P. As no quantization is involved, the bitrate is computed as $P \cdot 32$bit. The comparison between the use of Random Projections applied directly to the input fingerprint or residual ($\mathbf{r}$ dotted lines) against the use of Random Projections after decimation ($\mathbf{r}^*$ dashed lines) show that in the latter case the same TPR is obtained with a significant reduction of bitrate.

B. Quantization and coding

Figure 7 shows the reduction in terms of bitrate at equal TPR when dead-zone quantization is used instead of binarization for compressing query residuals. Camera fingerprints extracted from the $\mathcal{C}_f^{\mathrm{RAW}}$ database are decimated with $d = 2$, projected with Random Projection with $P = 96k$ and binarized. Query residuals from $\mathcal{Q}_n^{\mathrm{RAW}}$, $\mathcal{Q}_n^{\mathrm{QF=95}}$ and $\mathcal{Q}_n^{\mathrm{QF=90}}$ are first decimated with $d = 2$ then projected with Random Projection with varying P and binarized ($\mathbf{r}^b$ dotted lines) or projected with $P = 96k$ and quantized with a varying δ dead-zone quantizer ($\mathbf{r}^\delta$ dashed lines). The reported results show how for both uncompressed and JPEG compressed query images, the same TPR can be obtained with a bitrate reduction of more than 20% when using dead-zone quantization.

To confirm the choice of a dead-zone quantizer whose dead-zone is driven by the standard deviation σ of the residual, as described in Section IV, we also tested several different quantizers followed by an entropy coder and reported the results in Table I. We evaluate the required query rate to reach a TPR of 95%. In the first two lines, a Random Projection

with $96k$ output coefficients is fed to two different dead-zone quantizers, the top one with a σ-driven dead-zone and the second one with a signal independent dead-zone. In both cases, the values for δ are the same for all camera devices and a varying value of δ is used to draw a ROC curve. From the ROC curve we derive the bitrate needed to reach a 95% TPR. The other lines of the table are obtained by projecting the decimated residuals with a varying projection length P while quantizing with binarization, three uniform scalar quantizers and three Lloyd-Max scalar quantizers. The overall results from the table confirm the choice of a signal-dependent dead-zone quantizer as it reduces the required bitrate for fixed TPR performance.

As final step of the pipeline, the choice of a proper encoding scheme is essential to exploit the reduced entropy resulting from the dead-zone quantization. While results reported in the plots are computed with the use of a real arithmetic encoder, Table II shows the comparison between an arithmetic coder (AC) applied after dead-zone quantization (first row) compared to run-length coding (RLC) after dead-zone quantization (second row). Run-length coding is obtained by encoding only differential positions and sign of the peaks. The increased bitrate when using RLC is in any case smaller or equal to the bitrate obtained with binarized Random Projections applied to the original query residual (third row of Table I). In applications where bitrate constraints are relaxed, the choice of a run-length encoder allows to keep coding complexity at bay while preserving state of the art compression rates.

C. Query compression scenario

In a query compression scenario we wish to evaluate the trade-off between query residual bitrate and achieved True-Positive Rate. Three different datasets combinations are taken into account, all resorting to camera fingerprints from $\mathcal{C}_f^{\mathrm{RAW}}$ while query residuals are drawn from $\mathcal{Q}_n^{\mathrm{RAW}}$ (Figure 8), $\mathcal{Q}_n^{\mathrm{QF=95}}$ (Figure 9) and $\mathcal{Q}_n^{\mathrm{QF=90}}$ (Figure 10).

Each plot reports four curves comparing different methods: i) central fingerprint and residual cropping ($\hat{\mathbf{K}}$) while varying the amount of preserved pixels. Query residual coefficients are quantized by binarization; ii) Random Projections ($\mathbf{r}$) applied to the entire $\hat{\mathbf{K}}$ fingerprint while varying the number of projection components P. Projected coefficients from query residuals are quantized by binarization; iii) "Sub-Wrapping" method introduced in [38] (SW), which proved to behave at par with Random Projections with a lower computational complexity; iv) proposed method, with $\hat{\mathbf{K}}$ resized by a factor $d = 2$ that is then projected through Random Projections with $P = 96000$, giving rise to $\mathbf{r}_{96k}$. Query projected fingerprints are then quantized with a Dead-Zone quantizer ($\mathbf{r}^\delta$), where threshold δ is gradually increased to decrease query residual bitrate, thanks to arithmetic coding exploiting the reduced entropy of quantized residual.

By observing the three plots we can clearly see that when query residuals are extracted from uncompressed images (Figure 8) the performance gap between Random Projections applied to the entire fingerprint ($\mathbf{r}$) and dead-zone quantized Random Projections applied to resized fingerprints ($\mathbf{r}^\delta$) is negligible. When JPEG compression is applied to query images

TABLE I: Entropy coded query rate [kbit] @ TPR = 95% with several quantizer choices in joint and query compression scenarios. Best results in bold font.

P	Quantization	Joint QF 95	Joint QF 90	Query QF 95	Query QF 90
96k	Dead-zone, $\delta\sigma$	**22**	**30**	**13**	**18**
96k	Dead-zone, δ	44	54	35	42
varying	Binarization	30	39	19	26
varying	Uniform scalar, 3 levels	28	40	18	26
varying	Uniform scalar, 5 levels	33	45	23	31
varying	Uniform scalar, 7 levels	39	52	27	34
varying	Lloyd-Max scalar, 3 levels	34	46	22	33
varying	Lloyd-Max scalar, 5 levels	44	62	30	41
varying	Lloyd-Max scalar, 7 levels	53	68	34	46

TABLE II: Query rate [kbit] @ TPR = 95% with different encoders in joint and query compression scenarios. AC = Arithmetic Coding, RLC = Run-Length Coding

P	Quantization	Encoding	Joint QF 95	Joint QF 90	Query QF 95	Query QF 90
96k	Dead-zone	AC	22	30	13	18
96k	Dead-zone	RLC	28	40	18	23

(Figures 9 and 10) the gap between $\mathbf{r}$ and $\mathbf{r}^\delta$ increases greatly. If setting a goal TPR at around 95% the rate required by $\mathbf{r}$ is 25kbit and 52kbit, respectively for $\mathcal{Q}_n^{QF=95}$ and $\mathcal{Q}_n^{QF=90}$, while for $\mathbf{r}^\delta$ it is 13kbit and 18kbit, with a rate reduction between 48% and 65%.

To test performance of proposed method on a dataset with uncontrolled JPEG compression, Figure 11 reports results when camera fingerprints are from $\mathcal{C}_f^{JPG}$ and query residuals from $\mathcal{Q}_n^{JPG}$. In this case both camera and query images have undergone JPEG compression, but with several quality factors and customized quantization matrices, due to different brands' firmware implementations. In spite of the uncontrolled condition, when setting a desired TPR at 90% the proposed method achieves a 70% rate reduction with respect to Random Projections.

D. Joint database and query compression scenario

In a joint database and query compression scenario, experiments are carried out while quantizing by binarization all database fingerprints. Figures 12, 13 and 14 report results obtained respectively on query datasets $\mathcal{Q}_n^{RAW}$, $\mathcal{Q}_n^{QF=95}$ and $\mathcal{Q}_n^{QF=90}$. For each plot, the three lines represent performance with same query compression methods illustrated for query compression scenario. When query residuals are extracted from uncompressed images (Figure 12) the gap between $\mathbf{r}$ and $\mathbf{r}^\delta$ results negligible, while as soon as JPEG compression is applied to query images (Figures 13 and 14) the rate reduction obtained by $\mathbf{r}^\delta$ with respect to $\mathbf{r}$ is respectively of 46% and 66%, at a desired 95% TPR.

As previously done for the query compression scenario, also in the joint compression scenario we wish to verify performance under uncontrolled JPEG compression. Figure 15 reports results obtained from database fingerprints $\mathcal{C}_f^{JPG}$ and query residuals $\mathcal{Q}_n^{JPG}$. As for the query compression scenario, the rate reduction offered by the proposed method $\mathbf{r}^\delta$ with respect to Random Projection directly applied to $\hat{\mathbf{K}}$ is more than 68%, under a fixed 90% TPR.

Both in the query compression scenario (Figures 8, 9, 10) and in the joint compression scenario (Figures 12, 13, 14)

we can observe a common trend. The proposed method ($\mathbf{r}^\delta$) performs better in terms of compression than the state of the art method based on solely binarized Gaussian Random Projections ($\mathbf{r}$) when query images are JPEG compressed. However, when query images are uncompressed, the proposed method performs at par with the state of the art in terms of compression, but with a reduced computational complexity.

E. Lowering false-alarm probability

A last experiment is carried out by computing the Receiver-Operating-Characteristic (ROC) for both scenarios, query and joint compression, on the Dresden dataset. This test allows us to verify the performance of the proposed pipeline even at p_{FA} smaller than 0.05. With a fixed rate of 64kbit per query for all three compared compression methods, Figure 16 reports the obtained ROC curves for the query compression scenario, showing the Equal-Error-Rate for each curve at side of legend items. The same results are shown for the joint compression scenario in Figure 17. The proposed compression method preserves its good performance even at really small p_{FA}, making this choice viable also for those kind of systems that need to strictly bound the False Positive rate.

F. Running times

The execution time for the query compression pipeline is measured on a modern laptop equipped with a quad-core Intel Core-i7 processor on top of a MATLAB® 2018a implementation. The baseline pipeline that takes as input the image and directly applies Gaussian Random Projections followed by binarization and encoding takes 150ms. When decimation of a factor $d = 2$ is pre-pended to the same pipeline the running time drops to 38ms. Finally, as for the proposed method, when binarization is substituted by dead-zone quantization the total time required to execute the pipeline is 39ms.

VII. CONCLUSIONS

In this paper we presented a compression pipeline for PRNU fingerprints and residuals based on decimation, Random Projections and dead-zone quantization. At first we observed
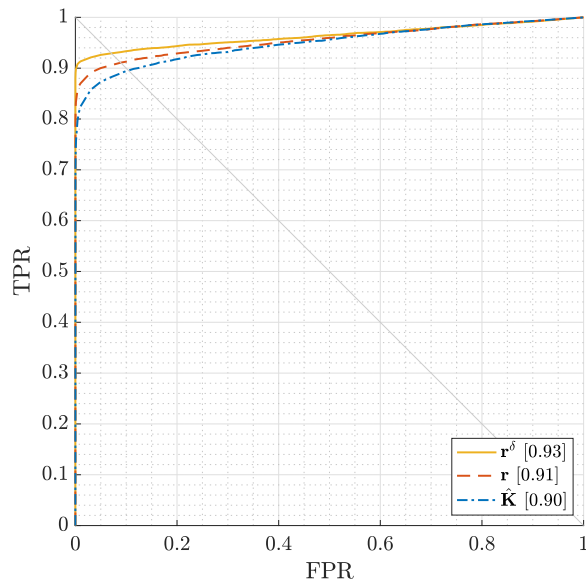
Fig. 16: Receiver-Operating-Characteristic at 64kbit per residual in a query compression scenario on Dresden dataset.
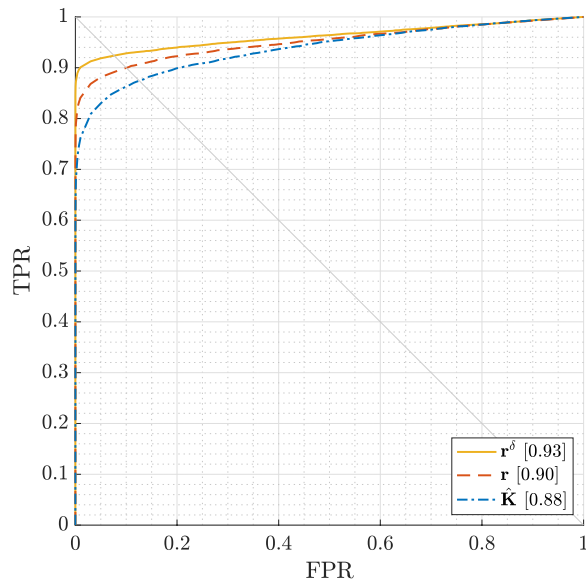

Fig. 17: Receiver-Operating-Characteristic at 64kbit per residual in a joint compression scenario on Dresden dataset.

that JPEG compression strongly attenuates high frequency components of the PRNU, basically zeroing the usefulness of such frequencies in terms of cross-correlation. Exploiting this phenomena, we decimate the extracted PRNU fingerprint and residuals before passing to Random Projections. Finally we are able to further reduce the bitrate by adopting a dead-zone quantization scheme, that fuses the advantages of fingerprint binarization and digest compression methods. On the Dresden Image Dataset, the proposed pipeline accounts for more than 65% bitrate reduction with respect to basic Random Projections applied to the whole fingerprint or residual, both in terms of query and joint compression, with an overall 75% complexity reduction.

REFERENCES

[1] A. Rocha, W. Scheirer, T. Boult, and S. Goldenstein, "Vision of the unseen: Current trends and challenges in digital image and video forensics," *ACM Comput. Surv.*, no. 4, pp. 26:1–26:42, 2011.

[2] M. C. Stamm, M. Wu, and K. J. R. Liu, "Information forensics: An overview of the first decade," *IEEE Access*, pp. 167–200, 2013.

[3] T. Gloe, M. Kirchner, A. Winkler, and R. Böhme, "Can we trust digital image forensics?" in *Proceedings of the 15th ACM International Conference on Multimedia*, ser. MM '07. New York, NY, USA: ACM, 2007, pp. 78–86.

[4] X. Feng, H. Zhang, H. C. Wu, and Y. Wu, "A new approach for optimal multiple watermarks injection," *IEEE Signal Processing Letters*, no. 10, pp. 575–578, 2011.

[5] J. Voisin, C. Guyeux, and J. M. Bahi, "The metadata anonymization toolkit," 2017, available at: https://mat.boum.org/.

[6] K. S. Choi, E. Y. Lam, and K. K. Y. Wong, "Source camera identification using footprints from lens aberration," in *Electronic Imaging*, 2006.

[7] L. T. Van, S. Emmanuel, and M. S. Kankanhalli, "Identifying source cell phone using chromatic aberration," in *Multimedia and Expo, 2007 IEEE International Conference on*. IEEE, 2007, pp. 883–886.

[8] S. Bayram, H. Sencar, N. Memon, and I. Avcibas, "Source camera identification based on CFA interpolation," in *IEEE International Conference on Image Processing 2005*. IEEE, 2005, pp. 69–72.

[9] A. Swaminathan, M. Wu, and K. Liu, "Nonintrusive component forensics of visual sensors using output images," *IEEE Transactions on Information Forensics and Security*, no. 1, pp. 91–106, 2007.

[10] E. J. Alles, Z. J. M. H. Geradts, and C. J. Veenman, "Source camera identification for heavily JPEG compressed low resolution still images," *Journal of Forensic Sciences*, no. 3, pp. 628–638, 2009.

[11] M. Kharrazi, H. Sencar, and N. Memon, "Blind source camera identification," in *2004 International Conference on Image Processing, 2004. ICIP '04*. IEEE, 2004, pp. 709–712.

[12] O. Celiktutan, B. Sankur, and I. Avcibas, "Blind identification of source cell-phone model," *IEEE Transactions on Information Forensics and Security*, no. 3, pp. 553–566, 2008.

[13] J. Lukas, J. Fridrich, and M. Goljan, "Determining digital image origin using sensor imperfections," in *Image and Video Communications and Processing 2005*, A. Said and J. G. Apostolopoulos, Eds. SPIE, 2005, p. 249.

[14] M. Goljan, M. Chen, and J. Fridrich, "Identifying common source digital camera from image pairs," in *2007 IEEE International Conference on Image Processing*, no. 1. IEEE, 2007, pp. VI – 125–VI – 128.

[15] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, "Determining image origin and integrity using sensor noise," *IEEE Transactions on Information Forensics and Security*, no. 1, pp. 74–90, 2008.

[16] J. Fridrich, "Digital image forensics," *IEEE Signal Processing Magazine*, no. 2, pp. 26–37, 2009.

[17] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, no. 2, pp. 205–214, 2006.

[18] I. Amerini, R. Caldelli, V. Cappellini, F. Picchioni, and A. Piva, "Analysis of denoising filters for photo response non uniformity noise extraction in source camera identification," in *2009 16th International Conference on Digital Signal Processing*. IEEE, 2009, pp. 1–7.

[19] G. Chierchia, S. Parrilli, G. Poggi, C. Sansone, and L. Verdoliva, "On the influence of denoising in prnu based forgery detection," in *Proceedings of the 2Nd ACM Workshop on Multimedia in Forensics, Security and Intelligence*, ser. MiFor '10. New York, NY, USA: ACM, 2010, pp. 117–122.

[20] A. Cortiana, V. Conotter, G. Boato, and F. De Natale, "Performance comparison of denoising filters for source camera identification," *Proceedings of the SPIE Conference on Media Watermarking, Security, and Forensics*, pp. 788 006–788 007, 2011.

[21] F. Gisolf, A. Malgoezar, T. Baar, and Z. Geradts, "Improving source camera identification using a simplified total variation based noise removal algorithm," *Digital Investigation*, no. 3, pp. 207–214, 2013.

[22] X. Lin and C.-T. Li, "Preprocessing reference sensor pattern noise via spectrum equalization," *IEEE Transactions on Information Forensics and Security*, no. 1, pp. 126–140, 2016.

[23] C. T. Li, "Source camera identification using enhanced sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 280–287, 2010.

[24] M. Goljan and J. Fridrich, "Camera identification from cropped and scaled images," in *Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, E. J. D. III, P. W. Wong, J. Dittmann, and N. D. Memon, Eds. SPIE, 2008, p. 68190E.

[25] K. Rosenfeld and H. T. Sencar, "A study of the robustness of PRNU-based camera identification," *Proc. SPIE 7254, Media Forensics and Security*, pp. 72 540M–72 540M–7, 2009.

[26] M. Goljan, J. Fridrich, and T. Filler, "Large scale test of sensor fingerprint camera identification," *Proceedings of SPIE*, no. 607, pp. 72 540I–72 540I–12, 2009.

[27] R. Caldelli, I. Amerini, F. Picchioni, and M. Innocenti, "Fast image clustering of unknown source images," in *2010 IEEE International Workshop on Information Forensics and Security*. IEEE, 2010.

[28] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva, "Blind PRNU-Based Image Clustering for Source Identification," *IEEE Transactions on Information Forensics and Security*, no. 9, pp. 2197–2211, 2017.

[29] D. Valsesia, G. Coluccia, T. Bianchi, and E. Magli, "User authentication via prnu-based physical unclonable functions," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1941–1956, Aug 2017.

[30] M. Goljan, J. Fridrich, and T. Filler, "Managing a large database of camera fingerprints," in *Media Forensics and Security II*, N. D. Memon, J. Dittmann, A. M. Alattar, and E. J. D. III, Eds. SPIE, 2010, p. 754108.

[31] S. Bayram, H. T. Sencar, and N. Memon, "Efficient sensor fingerprint matching through fingerprint binarization," *IEEE Transactions on Information Forensics and Security*, no. 4, pp. 1404–1413, 2012.

[32] Y. Hu, C.-T. Li, Z. Lai, and S. Zhang, "Fast camera fingerprint search algorithm for source camera identification," in *2012 5th International Symposium on Communications, Control and Signal Processing*, no. May. IEEE, 2012, pp. 1–5.

[33] Y. Hu, C.-T. Li, and Z. Lai, "Fast source camera identification using matching signs between query and reference fingerprints," *Multimedia Tools and Applications*, no. 18, pp. 7405–7428, 2014.

[34] F. Perez-Gonzalez, M. Masciopinto, I. Gonzalez-Iglesias, and P. Comesana, "Fast sequential forensic detection of camera fingerprint," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 3902–3906.

[35] D. Valsesia, G. Coluccia, T. Bianchi, and E. Magli, "Large-scale image retrieval based on compressed camera identification," *IEEE Transactions on Multimedia*, no. 9, pp. 1439–1449, 2015.

[36] ——, "Compressed fingerprint matching and camera identification via random projections," *IEEE Transactions on Information Forensics and Security*, no. 7, pp. 1472–1485, 2015.

[37] D. Valsesia and E. Magli, "Binary Adaptive Embeddings From Order Statistics of Random Projections," *IEEE Signal Processing Letters*, vol. 24, no. 1, pp. 111–115, Jan 2017.

[38] L. Bondi, F. Prez-Gonzlez, P. Bestagini, and S. Tubaro, "Design of projection matrices for prnu compression," in *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, Dec 2017, pp. 1–6.

[39] E. Quiring and M. Kirchner, "Fragile sensor fingerprint camera identification," in *2015 IEEE International Workshop on Information Forensics and Security*, 2015.

[40] M. Goljan, M. Chen, P. Comesaña, and J. Fridrich, "Effect of compression on sensor-fingerprint based camera identification," *Electronic Imaging*, no. 8, pp. 1–10, 2016.

[41] T. Gloe and R. Bhme, "The dresden image database for benchmarking digital image forensics," *Journal of Digital Forensic Practice*, no. 2-4, pp. 150–159, 2010.

[42] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1981.

Luca Bondi was born in Sassuolo, Italy, on February 12, 1990. He received the B.Sc. degree in Automation Engineering and the M.Sc. in Computer Science and Engineering from the Politecnico di Milano, Milan, Italy, in 2012 and 2014, respectively. He is currently a Ph.D. student at the Image and Sound Processing Group, Politecnico di Milano. His research interests focus on data-driven methods applied to images and videos in multimedia forensics applications.

Paolo Bestagini (M11) was born in Novara, Italy, on February 22, 1986. He received the M.Sc. degree in Telecommunications Engineering and the Ph.D. degree in Information Technology from the Politecnico di Milano, Italy, in 2010 and 2014, respectively. He is currently an Assistant Professor at the Image and Sound Processing Group, Politecnico di Milano. His research interests focus on multimedia forensics and acoustic signal processing for microphone arrays. He is an elected member of the IEEE Information Forensics and Security Technical Committee, and a co-organizer of the IEEE Signal Processing Cup 2018.

Fernando Pérez-González (M'90-SM'09-F'16) received the Telecommunication Engineer degree 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.

In 1990, he became an Assistant Professor with the School of Telecommunication Engineering, University of Vigo. From 2007 to 2010, he was Program Manager of the Spanish National R&D Plan on Electronic and Communication Technologies, Ministry of Science and Innovation. From 2009 to 2011, he was the Prince of Asturias Endowed Chair of Information Science and Technology, 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 the Principal Investigator of the University of Vigo Group, which participated in several European projects, including CERTIMARK, ECRYPT, REWIND, NIFTY, and WITDOM.

He is currently a Professor in the School of Telecommunication Engineering, University of Vigo, Vigo, Spain, and a Research Professor in Information Science and Technology, University of New Mexico, Albuquerque, NM, USA. He has coauthored more than 60 papers in leading international journals and 160 peer-reviewed conference papers. He has coauthored several international patents related to watermarking for video surveillance, integrity protection of printed documents, fingerprinting of audio signals, and digital terrestrial broadcasting systems. His research interests include the areas of digital communications, adaptive algorithms, privacy enhancing technologies, and information forensics and security. Prof. Pérez-González was an Associate Editor of the IEEE SIGNAL PROCESSING LETTERS (2005-2009) and the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (2006-2010). He is currently is an Associate Editor of the LNCS Transactions on Data Hiding and Multimedia Security, and Editor in Chief of the EURASIP Journal on Information Security.

Stefano Tubaro (SM01) was born in Novara, Italy, in 1957. He completed his studies in Electronic Engineering at the Politecnico di Milano, Milan, Italy, in 1982. He then joined the Dipartimento di Elettronica, Informazione e Bioingegneria of the Politecnico di Milano, first as a Researcher of the National Research Council, and then (in November 1991) as an Associate Professor. Since December 2004, he has been appointed as a Full Professor of telecommunication at the Politecnico di Milano. His current research interests include advanced algorithms for video and sound processing. He is the author of more than 180 scientific publications on international journals and congresses and the coauthor of more than 15 patents. In the past few years, he has focused his interest on the development of innovative techniques for image and video tampering detection and, in general, for the blind recovery of the processing history of multimedia objects. He coordinates the research activities of the Image and Sound Processing Group at the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano. He had the role of Project Coordinator of the European Project ORIGAMI: A new paradigm for high-quality mixing of real and virtual and of the research project ICT-FET-OPEN REWIND: REVerse engineering of audio-VIsual coNtent Data. This last project was aimed at synergistically combining principles of signal processing, machine learning, and information theory to answer relevant questions on the past history of such objects. He is a member the IEEE Multimedia Signal Processing Technical Committee and of the IEEE SPS Image Video and Multidimensional Signal Technical Committee. He was in the organization committee of a number of international conferences including the IEEE MMSP 2004/2013, IEEE ICIP 2005, IEEE AVSS 2005/2009, IEEE ICDSC 2009, IEEE MMSP 2013, IEEE ICME 2015. From May 2012 to April 2015, he was an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, and is currently an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY.