Pattern Recognition 74 (2018) 556–567

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Inference of a compact representation of sensor fingerprint for source camera identification

Ruizhe Li [a], Chang-Tsun Li [a,b,*], Yu Guan [c]

[a] Department of Computer Science, University of Warwick, UK
[b] School of Computing and Mathematics, Charles Sturt University, Australia
[c] Open Lab, School of Computing, Newcastle University, UK

ARTICLE INFO

Article history:
Received 30 November 2016
Revised 28 August 2017
Accepted 18 September 2017
Available online 27 September 2017

Keywords:
Image forensics
Source camera identification (SCI)
Sensor pattern noise (SPN)
PCA de-noising

ABSTRACT

Sensor pattern noise (SPN) is an inherent fingerprint of imaging devices, which provides an effective way for source camera identification (SCI). Although SPNs extracted from large image blocks usually yield high identification accuracy, their high dimensionality would incur a high computational cost in the matching stage, consequently hindering many applications that require efficient camera matchings. In this work, we employ and evaluate the concept of principal component analysis (PCA) de-noising in SCI tasks. Based on this concept, we present a framework that formulates a compact SPN representation. To enhance the de-noising effect, we introduce a training set construction procedure that minimizes the impact of various interfering artifacts, which is especially useful in some challenging cases, e.g., when only textured reference images are available. To further boost the SCI performance, a novel approach based on linear discriminant analysis (LDA) is adopted to extract more discriminant SPN features. To evaluate our methods, extensive experiments are conducted on the Dresden image database. The results indicate that the proposed framework can serve as an effective post-processing procedure, which not only boosts the performance, but also greatly reduces the computational cost in the matching phase.

1. Introduction

Nowadays, the use of digital images or videos as evidence in the fight against physical crime and cybercrime is a norm, which makes multimedia forensics crucial. Typically, multimedia forensics includes source camera verification and identification, source-oriented images classification, integrity verification, forgery detection, authentication, etc. Source camera identification, as an important branch of multimedia forensics, is about answering the question: *Which one of the many cameras has taken the image in question?* This is actually a task of matching the camera fingerprint of an image in question to a set of reference fingerprints, each representing a different camera. The size of the reference fingerprint set can be in the order of millions. How to deal with such a task more accurately and efficiently is the focus of this paper.

In order to link digital images to the source cameras, many techniques have been proposed in the last two decades. These techniques can be broadly divided into three categories. The simplest way is to use digital images' metadata that contains the information of the source camera [1].

However, due to the wide prevalence and great user-friendliness of multimedia processing tools nowadays (e.g., Adobe Photoshop and IrfanView), metadata can be easily changed or removed by laymen. Therefore, metadata is no longer regarded as reliable for authentication purposes. Another possible way is to use the digital watermark, which is a signature embedded in the image by a certain type of cameras [2]. This technique is useful in the cases of proving ownership of copyright. Yet it is only applicable to the cameras that have watermarking mechanism [2]. The third category of techniques rely on the intrinsic characteristics of digital cameras left in the captured images. Many traces left in the content by various hardware and software components in the image acquisition pipeline can be exploited to link the image to its source camera. Good examples are sensor pattern noise (SPN) [3–8], lens aberrations [9], color filter array (CFA) interpolation artefacts [10], JPEG compression [1], and the combination of several intrinsic characteristics [11]. Among these modalities, SPN has been proved to be the most effective camera fingerprint as it is capable of differentiating individual cameras of the same model.

Sensor pattern noise is produced by the imaging sensor and primarily caused by the manufacturing imperfections and the in-

* Corresponding author.
E-mail addresses: ruizhe.li@warwick.ac.uk (R. Li), c-t.li@warwick.ac.uk (C.-T. Li), yu.Guan@ncl.ac.uk (Y. Guan).

https://doi.org/10.1016/j.patcog.2017.09.027

homogeneity of silicon wafers. It is essentially the slight variations in the intensity of individual pixels. For instance, even if a sensor takes an image of an evenly lit scene, the resulting image will still exhibit slight changes in intensity between individual pixels [3]. Every image taken by the same sensor would exhibit the same SPN pattern, while two sensors, even made from the same silicon wafer, would exhibit uncorrelated patterns [3].

The dimensionality of SPN is as large as that of the original image. As a result, not only each SPN needs a fairly large amount of space for storage, but memory access would also take considerable amount of time. Moreover, SPN matching involves vector operations and the complexity is proportional to the size of SPNs. Thus, with a large number of reference SPN in the database to be matched, the complexity of matching process would become a critical concern.

In order to address the high complexity issue, many efforts [12–18] have been made in recent years. In [12], Bayram *et al.* embeded reference SPNs in a binary search tree, where the leaf/internal node represents a reference/composite SPN. Based on this structure, the total number of SPN matchings to be performed is substantially reduced. However, errors tend to increase significantly when a large number of reference SPNs are stored in a single binary tree. On the other hand, more methods reduce the computational complexity by compressing the SPN. In [13,14], the authors introduced a SPN digest technique for dimensionality reduction, which preserves the largest elements and their corresponding locations. In [15], Bayram *et al.* binarized SPN, which considerably reduces the storage requirements and speeds up loading of SPN into the memory. However, the binarization process inevitably degrades the matching accuracy due to information loss. In [16,17], Valsesia *et al.* reduced the dimensionality of SPN using random projection. However, since the subspace is randomly selected, the obtained representation is unlikely to be optimal and tends to compromise the matching accuracy.

To alleviate the common limitation (i.e., reduced accuracy) of the afore-mentioned SPN compression methods [13–17], in our previous work [19,20], we presented a feature extraction algorithm based on the concept of PCA de-noising [21,22], and promising results were achieved on a small dataset. However, this method is based on the assumption that the training set is well representative of the population so that an effective SPN feature extractor can be learned. Unfortunately, the noise residuals in the training set can be contaminated by many sources of interference, making the training set less representative. To learn a robust SPN feature extractor from the noisy training data, in this work, we further propose a training set construction procedure and provide its theoretical basis. We also provide more detailed discussion of the SPN feature extractors and treat it as a general post-processing framework on other SPN methods. It is evaluated in term of effectiveness and efficiency on a much larger dataset. We also test this framework on some challenging cases, e.g., all the reference SPNs are extracted from images with significant scene details (a form of distortion to the SPN), which are scenarios barely considered by previous works.

The rest of this paper is organized as follows. Section 2 provides a brief review on the three main steps of the SPN-based SCI system. In Section 3, we present the proposed training dataset construction procedure and the feature extraction method in details. In Section 4, the proposed source camera identification method is summarized, which is then followed by extensive experimental evaluations in Section 5. Section 6 concludes the work. Note that, in this manuscript, we use bold upper-case letters to represent matrices, and bold lower-case letters to denote vectors.

2. Background

In order to decide whether a query image is taken by one of the cameras in a large dataset, three main steps are required, i.e., SPN extraction, reference SPN estimation and SPN matching. In this section, techniques for these three steps are briefly reviewed.

2.1. SPN extraction

The most important step of the SPN-based SCI framework is to extract the SPNs from digital images. In [4], Chen *et al.* modeled the output of imaging sensor $\mathbf{I}$ and explained the general idea about how to extract SPN, such as

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

In Eq. (1), $\mathbf{I}^{(0)}$ is the noiseless sensor output and $\mathbf{I}^{(0)}\mathbf{K}$ represents the discriminative part of SPN, i.e., PRNU noise, which is a multiplicative noise and the signal of our interest. The matrix $\mathbf{K}$ is the PRNU multiplicative factor, where all the elements in it are typically close to 0. $\boldsymbol{\Theta}$ is a combination of random noise, such as shot noise, read-out noise, and quantization noise. In order to extract the signal of interest $\mathbf{I}^{(0)}\mathbf{K}$ from the observation $\mathbf{I}$, the host signal $\mathbf{I}^{(0)}$ should be removed. Generally, the noiseless image $\mathbf{I}^{(0)}$ is unknown, but we can estimate it by de-noising the observation $\mathbf{I}$, i.e., $\hat{\mathbf{I}}^{(0)} = F(\mathbf{I})$, where F indicates a de-noising algorithm and $\hat{\mathbf{I}}^{(0)}$ is an estimation of the noiseless image $\mathbf{I}^{(0)}$. Then, the signal of interest can be roughly extracted by subtracting the estimation $\hat{\mathbf{I}}^{(0)}$ from the observation $\mathbf{I}$, such as

$$\begin{aligned}\mathbf{X} &= \mathbf{I} - F(\mathbf{I}) = \mathbf{I} - \hat{\mathbf{I}}^{(0)} \\ &= \mathbf{I}^{(0)} + \mathbf{I}^{(0)}\mathbf{K} + \boldsymbol{\Theta} - \hat{\mathbf{I}}^{(0)} \\ &= \mathbf{I}\mathbf{K} + \mathbf{I}^{(0)} - \hat{\mathbf{I}}^{(0)} + (\mathbf{I}^{(0)} - \mathbf{I})\mathbf{K} + \boldsymbol{\Theta} \\ &= \mathbf{I}\mathbf{K} + \boldsymbol{\Xi}\end{aligned} \tag{2}$$

where $\mathbf{X}$ is the noise residual where the true SPN is present, $\boldsymbol{\Xi}$ is the sum of $\boldsymbol{\Theta}$ and two additional noise terms introduced by the de-noising filter.

From Eq. (2), one can see that the better a de-noising algorithm F is, the closer the de-noised version $\hat{\mathbf{I}}^{(0)}$ is to the noiseless image $\mathbf{I}^{(0)}$, and thus the less noise would be introduced by the de-noising filter and left in the final output $\mathbf{X}$. Therefore, the performance of a SPN extractor is primarily determined by the choice of the de-noising algorithm F. In [3], Lukas *et al.* proposed to transform the noisy image $\mathbf{I}$ into wavelet transform domain and apply the Mihcak filter [23] to extract the SPN components from the high frequency wavelet coefficients of $\mathbf{I}$. In [24], Chierchia *et al.* proposed to replace the Mihcak filter with a more recent technique, namely the sparse 3D transform-domain collaborative filtering [25]. In [26], Kang *et al.* proposed a SPN predictor based on context adaptive interpolation (PCAI), which is to apply the context adaptive interpolator [27] as the de-noising function F to predict the noiseless image $\mathbf{I}^{(0)}$ and extract SPN in the spatial domain.

Also demonstrated in Eq. (2) is the fact that the noise residual $\mathbf{X}$ contains not only the SPN term $\mathbf{I}\mathbf{K}$ but also the noise term $\boldsymbol{\Xi}$. This leaves room for further enhancement. In [5], Li demonstrated that the noise residual contains the traces of scene details. Therefore, Li proposed 5 enhancing models to attenuate the impact of scene details. In [28], Li and Li proposed a color-decoupled SPN extraction method to prevent the color interpolation errors from propagating into the noise residual. In [29], Chen *et al.* proposed to suppress the JPEG blocky artifacts by transforming the noise residual into the discrete Fourier transform domain and suppressing the Fourier coefficients with extremely larger magnitude.

2.2. Reference SPN estimation

This step aims at estimating the reference SPN for a camera. Typically, the reference SPN, **R**, for a camera is estimated by averaging N (e.g., $N \geq 20$) noise residuals extracted from flat-field/low-variation images (e.g., blue sky images) taken by that camera, such as

$$\mathbf{R} = \sum_{i=1}^{N} \mathbf{X}_i / N \tag{3}$$

Random noise presented in different images are different, while the true SPN components would be the same as long as these images are taken by the same camera. Therefore, the random noise components can be averaged out in **R** while the true SPN components are accumulated. In [4], Chen *et al.* proposed a maximum likelihood estimation (MLE) method to estimate the reference SPN. They also proposed two enhancing operations, namely zero-mean (ZM) and Wiener filtering (WF) in the discrete Fourier transform (DFT) domain, to remove the artifacts caused by camera processing operations from the reference SPN. In [30], Lin and Li argued that the true SPN is unlikely to be periodic and should have a flat spectrum. Therefore, they proposed another reference enhancing method, namely spectrum equalization algorithm (SEA), to detect and suppress the peaks appearing in the DFT spectrum of the reference SPN so as to remove the periodic artifacts.

2.3. SPN matching

Once both query SPN and reference SPN are obtained, the matching step can be performed. Such a task can be treated as a binary hypothesis test as follows

H_0 : $\mathbf{X} \neq \mathbf{R}_i$ (the query image is not taken by the ith camera).

H_1 : $\mathbf{X} = \mathbf{R}_i$ (the query image is taken by the ith camera).

Here a correlation-based detector is used to make the decision between H_0 and H_1 by comparing the correlation $\rho(\mathbf{X}, \mathbf{R}_i)$ to a pre-calculated threshold τ. The detector accepts H_1 when $\rho \geq \tau$ or H_0 when $\rho < \tau$. The normalized cross-correlation (NCC) is usually used to measure the similarity between the query noise residual $\mathbf{X} \in \mathbb{R}^{M \times M}$ and the reference SPN $\mathbf{R} \in \mathbb{R}^{M \times M}$, which is defined as

$$\rho(\mathbf{X}, \mathbf{R}) = \frac{\sum_{i=1}^{M} \sum_{j=1}^{M} \left(\mathbf{X}[i,j] - \overline{\mathbf{X}}\right)\left(\mathbf{R}[i,j] - \overline{\mathbf{R}}\right)}{\left\|\mathbf{X} - \overline{\mathbf{X}}\right\| \cdot \left\|\mathbf{R} - \overline{\mathbf{R}}\right\|}, \tag{4}$$

where $\overline{\mathbf{X}}$ and $\overline{\mathbf{R}}$ are the mean value of **X** and **R**, and $\|\cdot\|$ is the L_2 norm. Given an upper bound on the false positive rate (FPR), the threshold τ for the detector can be calculated via the Neyman–Pearson approach [31]. In [32], Goljan pointed out that NCC is sensitive to the influence of periodic noise, and proposed the peak-to-correlation energy (PCE) ratio as a replacement to measure the similarity between two SPNs. More recently, Kang *et al.* [6] proposed another measurement, namely correlation over circular correlation norm, to reduce the FPR of a SCI system.

The aforementioned methods can be combined to further boost performance gains. For example, forensic investigators can apply the Mihcak filter to extract the noise residuals from both query and reference images, and enhance the query noise residuals with Li's enhancing models [5] and improve the reference SPNs with either the ZM+WF operations [4] or the SEA algorithm [30], and finally apply NCC or PCE as the similarity measurement for SPN matching. Moreover, in many applications, such as source-oriented image clustering and SCI in a large-scale reference SPN databases, taking the full-sized image into account is not computationally feasible and a block smaller than the full-sized image is used. Due to the vignetting effect on the peripherals of images [33], it is suggested that such a block is better cropped from the center of the full-sized image. The noise residuals extracted from larger blocks usually yield higher identification accuracy, but they also have high dimensionality. The complexity of matching a query image with the camera in the database is $O(mc)$, where m is the dimensionality of each noise residual and c is the number of cameras in the database. Considering the fact that there may be tens of thousands of reference SPNs (each representing a camera) in the database, matching the high dimensional noise residuals may incur excessive computational costs. To address this problem, we propose a new approach in the next section.

3. Proposed SPN feature extraction and enhancement

Generally speaking, high-dimensional SPNs not only incur a high computational costs but also tend to contain more redundancy and interfering components. For simplicity, we write Eq. (2) as the sum of the true SPN and unwanted noise, i.e.,

$$\mathbf{X} = \mathbf{X}^{(0)} + \Xi, \tag{5}$$

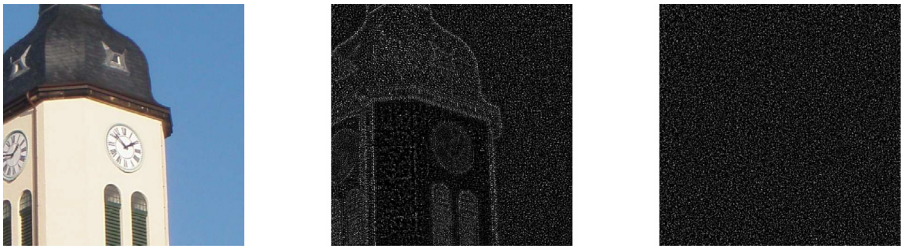
where $\mathbf{X}^{(0)}$ is the true SPN, and Ξ represents an additive mixture of unwanted interferences, which may include scene details and the artifacts introduced by color interpolation, JPEG compression and other camera processing operations [4]. The former can be scene-specific, while the latter can be shared among cameras of the same model or sensor design. Therefore they are non-unique, less discriminant and redundant. In order to improve the performance of SCI systems, one intuitive way is to suppress these artefacts Ξ.

PCA [34] is a well-known unsupervised learning method, which minimizes the reconstruction error using a linear transformation, and can be used to learn compact representation of high-dimension data. This method has been widely used for the purpose of de-noising [21,22], dimensionality reduction [35], feature extraction [36], etc. Compared with data-independent dimensionality reduction methods, such as random projection, the PCA projection matrix is learned based on a training data, and it generally has higher performance in classification tasks [37]. In this work, we attempt to find a PCA transformed domain, where the true SPN is well represented. Ideally, by projecting the extracted noise residuals onto this domain, a small set of coefficients that contain most of the representative information of the true SPN can be extracted.

3.1. Training set construction

In order to identify such a transformed domain, a representative training set needs to be established in advance. PCA is to find an optimal transformed domain that better represents the primary signal shared among the training samples. So if SPN appears as the most representative signal among the training samples, it would be better to represent it in the obtained domain. However, some contamination (e.g., scene details) can be more dominant than SPN in the noise residual (as shown in Fig. 1(b)). Without removing these strong contaminations from the training set, the obtained domain is more likely to represent these noisy components rather than the true SPN. To avoid this situation, we propose the following strategies to minimize the impact of the unwanted noise in the training set:

1. *Training sample selection:* To build the training set, if we have access to the cameras in the database, we give the priority to the noise residuals extracted from flat-field images (e.g., blue sky). Such images are more similar to the evenly lit scene and contain less scene details so that these images can better exhibit the changes caused by SPN. However, in many real-world scenarios, the cameras in question may not be in the investigator's possession, making it impossible for the investigator to use



(a) (b) (c)

Fig. 1. (a) An image taken by Olympus_mju_1050SW. (b) The noise residual extracted from (a) using the Mihcak de-noising filter. (c) The reconstructed version of (b). Note the intensity of (b) and (c) has been down scaled 5 times and up scaled 2 times, respectively, for visualization purpose.

the cameras to take flat-field images. Instead only images with varying scene details taken by those cameras are available (e.g., from someone's Facebook account). In this case, our strategy is to suppress the impact of scene details through averaging. Considering the fact that scene details presented in different images are normally different, we can generate a smoother sample by averaging several noise residuals of the images taken by the same camera. By repeating this process several times, we can finally generate a set of training samples, which are more representative.

We also model the afore-mentioned contamination-removal process based on Eq. (5). In this context, $\boldsymbol{\theta}$ represents the scene details, while $\hat{\mathbf{X}}$ is the sum of SPN and some non-unique artifacts (e.g., CFA pattern and JPEG blocky artifacts), which will not be suppressed by averaging in this stage. Given that, for a camera with N reference images, each pixel's mean and variance in the reference SPN can be expressed as $\mu_{\mathbf{X}} = \hat{\mathbf{X}} + \frac{1}{N}\sum_i^N \boldsymbol{\theta}_i$, and $\sigma_{\mathbf{X}}^2 = \mathbb{E}[(\boldsymbol{\theta}_i - \frac{1}{N}\sum_i^N \boldsymbol{\theta}_i)^2]$, $i = 1, 2, \ldots, N$, respectively. For a camera, if we repeat averaging the SPNs of a random subset of T out of the N reference images for L times, then according to Eq. (5) we will have

$$\mathbf{X}'_l = \hat{\mathbf{X}} + \frac{1}{T}\sum_t^T \boldsymbol{\theta}_{lt}, \quad l = 1, 2, \ldots, L. \tag{6}$$

The new mean and variance for each pixel can be expressed as follows

$$\mu_{\mathbf{X}'} = \hat{\mathbf{X}} + \frac{1}{LT}\sum_l^L\sum_t^T \boldsymbol{\theta}_{lt}, \tag{7}$$

$$\sigma_{\mathbf{X}'}^2 = \frac{1}{L}\sum_l^L \left(\frac{1}{T}\sum_t^T \boldsymbol{\theta}_{lt} - \frac{1}{LT}\sum_l^L\sum_t^T \boldsymbol{\theta}_{lt}\right)^2. \tag{8}$$

In Eq. (8), the term $\frac{1}{LT}\sum_l^L\sum_t^T \boldsymbol{\theta}_{lt}$ can be approximated as the mean of the scene details $\bar{\boldsymbol{\theta}} = \frac{1}{N}\sum_i^N \boldsymbol{\theta}_i$ when the product of T and L is large. For simplicity, in this work we set $L = N$ to generate as many samples as the original data. In this case, if we set $T \to N$, the term $\frac{1}{T}\sum_t^T \boldsymbol{\theta}_{lt}$ of Eq. (8) also converges to the mean of the scene details $\bar{\boldsymbol{\theta}} = \frac{1}{N}\sum_i^N \boldsymbol{\theta}_i$, which makes $\sigma_{\mathbf{X}'}^2 \to 0$, hence suppressing the interference of scene details.

2. *Training sample enhancement:* In addition to scene details, some non-unique artifacts such as CFA patterns and JPEG blocky artifacts may also lead to unsatisfactory training. Since these artifacts in the images taken by the cameras of the same model or brand are similar (with small variation), they would survive the averaging operation. Nevertheless, as we have shown in [30], these artifacts cause peaks in the DFT magnitude spectrum, while the SPN appears as a flat spectrum without salient peaks. Therefore, by suppressing the peaks present in the DFT spectrum, these artifacts can be effectively suppressed and the quality of the true SPN in the noise residual can be thereby enhanced.

Assume there are n reference images $\{\{\mathbf{I}_{ij}\}_{i=1}^N\}_{j=1}^c$ taken by c cameras $\{C_j\}_{j=1}^c$, each responsible for N images such that $n = cN$. According to the two afore-mentioned strategies for training sample selection and enhancement, we can summarize the proposed training set construction as follows:

(1) Extract the 2D noise residuals $\{\{\mathbf{X}_{ij}\}_{i=1}^N\}_{j=1}^c$ from the blocks of $W \times W$ pixels cropped from the center of the n reference images.
(2) For each camera C_j, randomly select T noise residuals from $\{\mathbf{X}_{ij}\}_{i=1}^N$ (belong to camera C_j) for averaging.
(3) Detect and suppress the peaks of the averaged noise residual in the DFT magnitude specturm with SEA [30]. Then concatenate the 2D output into a column vector as a training sample $\mathbf{x}_{ij}$. Note that we use $\mathbf{X}_{ij}$ to represent 2D noise residuals and $\mathbf{x}_{ij}$ to represent their 1-d version.
(4) Repeat the process in Steps (2) and (3) L times for each camera to form the training set $\{\{\mathbf{x}_{ij}\}_{i=1}^L\}_{j=1}^c \in \mathbb{R}^m$, where $m = W \times W$.

In Step (2), we randomly select T images from each camera for averaging. As discussed above, it is preferable to set T to a larger value so as to better attenuate the impact of scene details and random noise. However, since the CFA pattern and JPEG blocky artifacts are shared among the images taken by the camera, the averaging operation would also inevitably enhance these two artifacts in each training sample. However, the peaks caused by these artifacts are more distinct in the DFT spectrum and they can be more easily and accurately detected. Given that, setting T to a large value would also help SEA to achieve a more accurate peak detection in Step (3), which would consequently increase the effect of enhancement. More details about how the setting of T affects the performance is discussed in Section 5.2.

3.2. SPN feature extraction through PCA

PCA is performed to seek a set of orthonormal eigenvectors $\{\mathbf{v}_k\}_{k=1}^m$ and their associated eigenvalues $\{\lambda_k\}_{k=1}^m$ of the covariance matrix $\mathbf{S}$ given by

$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T = \mathbf{A}\mathbf{A}^T, \tag{9}$$

where $\mathbf{A} = \frac{1}{\sqrt{n}}[\mathbf{x}_1 - \bar{\mathbf{x}}, \ldots, \mathbf{x}_n - \bar{\mathbf{x}}] \in \mathbb{R}^{m\times n}$ and $\bar{\mathbf{x}}$ is the global mean estimated by $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$. The eigenvectors $\mathbf{v}_k$ and eigenvalues λ_k are obtained by solving the eigenvalue decomposition $\mathbf{S}\mathbf{v}_k = \lambda_k\mathbf{v}_k$, in which $k = 1, \ldots, m$. Given that the dimensionality of SPNs can be extremely high (e.g., $m > 10^7$), directly decomposing $\mathbf{S} \in \mathbb{R}^{m\times m}$ would incur a prohibitively high computational cost (with a complexity $O(m^3)$). To make PCA feasible for the high-dimensional SPN,

we apply a fast method [38] instead of computing these eigenvectors when $m \gg n$.

Assuming $\mathbf{v}_k{}'$ is the unit eigenvector of $\mathbf{A}^T\mathbf{A} \in \mathbb{R}^{n\times n}$ with eigenvalue λ_k', we can obtain $\mathbf{A}^T\mathbf{A}\mathbf{v}_k{}' = \lambda_k{}'\mathbf{v}_k{}'$. By multiplying both sides by $\mathbf{A}$, we get

$$\mathbf{A}\mathbf{A}^T(\mathbf{A}\mathbf{v}_k{}') = \lambda_k{}'(\mathbf{A}\mathbf{v}_k{}'), \tag{10}$$

where $\mathbf{A}\mathbf{v}_k{}'$ are the eigenvectors of $\mathbf{A}\mathbf{A}^T = \mathbf{S}$ with eigenvalues $\lambda_k{}'$. Thus, instead of decomposing matrix $\mathbf{S}$ directly, we can calculate the eigenvectors $\mathbf{v}_k{}'$ by decomposing a smaller matrix $\mathbf{A}^T\mathbf{A} \in \mathbb{R}^{n\times n}$. Then $\mathbf{v}_k$ can be obtained via $\mathbf{v}_k = \mathbf{A}\mathbf{v}_k{}'$. Computing eigenvectors in such a manner incurs a complexity of $O(n^3)$. Considering the fact that the number of training samples tends to be much smaller than the size of SPNs (i.e., $n \ll m$), thus computing eigenvectors in such a manner would be more efficient than the traditional one. The obtained $\{\mathbf{v}_k\}_{k=1}^n$ are normalized and sorted in the descending order according to their corresponding eigenvalues $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \lambda_n$. Subsequently, a transformed domain can be built as $\mathbf{M}_{pca} = [\mathbf{v}_1, \ldots, \mathbf{v}_n] \in \mathbb{R}^{m\times n}$. After that, we can apply $\mathbf{M}_{pca}$ to noise residual $\mathbf{x}$ (defined in Eq. (5)) through

$$\begin{aligned}\mathbf{y} &= \mathbf{M}_{pca}^T\mathbf{x} = \mathbf{M}_{pca}^T(\mathbf{x}^{(0)} + \boldsymbol{\Xi}) \\ &= \mathbf{M}_{pca}^T\mathbf{x}^{(0)} + \mathbf{M}_{pca}^T\boldsymbol{\Xi} = \mathbf{y}^{(0)} + \boldsymbol{\Xi}_{\mathbf{y}},\end{aligned} \tag{11}$$

where $\mathbf{y}^{(0)}$ and $\boldsymbol{\Xi}_{\mathbf{y}}$ are the transformed versions of the SPN term and the noise term, respectively. Now the problem is recast as estimating $\mathbf{y}^{(0)}$ from the noisy $\mathbf{y}$. Generally speaking, in a PCA transformed vector (i.e., $\mathbf{y}$), most energy of the primary signal among the training set would concentrate on the first several elements of $\mathbf{y}$, while the energy of the noise would be distributed in $\mathbf{y}$ much more evenly. Therefore, only retaining the first several elements of $\mathbf{y}$ while discarding the rest would preserve the energy of the signal of interest and suppress the energy of the noise. Following this concept, the eigenvectors with the d largest eigenvalues are selected to form an *SPN feature extractor* $\mathbf{M}_{pca}^d = [\mathbf{v}_1, \ldots, \mathbf{v}_d] \in \mathbb{R}^{m\times d}$, with d satisfying

$$d = \min\{d' \mid \textstyle\sum_{i=1}^{d'} \lambda_i / \sum_{i=1}^{n} \lambda_i > 98\%\}. \tag{12}$$

With this SPN feature extractor $\mathbf{M}_{pca}^d$, we can obtain a new feature with much lower dimensionality by

$$\begin{aligned}\mathbf{y}^d &= (\mathbf{M}_{pca}^d)^T\mathbf{x} = (\mathbf{M}_{pca}^d)^T\mathbf{x}^{(0)} + (\mathbf{M}_{pca}^d)^T\boldsymbol{\Xi} \\ &= \mathbf{y}^{(0)^d} + \boldsymbol{\Xi}_{\mathbf{y}}^d,\end{aligned} \tag{13}$$

where $\mathbf{y}^d$ is the compact representation of $\mathbf{y}$. With the feature vector $\mathbf{y}^d$ and SPN feature extractor $\mathbf{M}_{pca}^d$, it is reasonable to assume that we can obtain a reconstructed SPN in the spatial domain via the inverse PCA transform as follow

$$\mathbf{x}' = (\mathbf{M}_{pca}^d)\mathbf{y}^d, \tag{14}$$

where $\mathbf{x}'$ is an approximation of the original $\mathbf{x}$. If our assumption is correct, noise $\boldsymbol{\Xi}_{\mathbf{y}}$ should be suppressed by the PCA-based SPN feature extractor. As a consequence, the reconstructed $\mathbf{x}'$ should contain less noise and have a higher signal-to-noise ratio (SNR) than the original noise residual $\mathbf{x}$. To validate our assumption, we demonstrate the behavior of our SPN feature extractor with a simple example. As shown in Fig. 1(b), the scene details in Fig. 1(a) propagates through the Wiener filter into the noise residual. After performing the proposed SPN feature extraction and inverting the PCA transformation, the artifacts caused by the scene details have been significantly suppressed in the reconstructed SPN, as shown in Fig. 1(c). The effect of the proposed method can also be quantitatively evaluated by comparing the Signal-to-Noise Ratio (SNR) of the true SPN, to the contaminated version (Fig. 1(b)) and (to the reconstructed SPN (Fig. 1(c)). First, the true SPN $\mathbf{x}^{(0)}$ is estimated by averaging 50 noise residuals extracted from blue sky images. According to Eq. (5), the noise $\boldsymbol{\Xi}$ in the noise residual (Fig. 1(b)) and (the reconstructed SPN (Fig. 1(c)) can be estimated by subtracting the true SPN $\mathbf{x}^{(0)}$ from the observed data, respectively. Then, the SNR can be calculated according to $10\log_{10}\frac{var(\mathbf{x}^{(0)})}{var(\boldsymbol{\Xi})}$. As expected, the reconstructed SPN has a much higher average SNR (4.3 dB) than the original noise residual (−15.5 dB), which further validates our assumption.

3.3. SPN feature enhancement through LDA

In the task of SCI, the source cameras of the images in the database are usually known, which means the class label of each image is known. If this is the case, by taking advantage of this prior knowledge, we can further extract a more discriminant feature by using a supervised learning method, i.e., linear discriminant analysis (LDA) [39]. The purpose of using LDA in this work is to build an enhancer $\mathbf{M}_{lda}$ to enhance the SPN feature extractor $\mathbf{M}_{pca}$ so as to extract more compact representation from the original noise residual $\mathbf{x}$. This enhancer can be obtained by maximizing the ratio of the determinant of the between-class scatter matrix $\mathbf{S}_b$ to the determinant of the within-class scatter matrix $\mathbf{S}_w$

$$\mathbf{M}_{lda} = \hat{\mathbf{J}} = \arg\max_{\mathbf{J}} \left|\frac{\mathbf{J}^T\mathbf{S}_b\mathbf{J}}{\mathbf{J}^T\mathbf{S}_w\mathbf{J}}\right|, \tag{15}$$

where $\mathbf{S}_w$ is defined as $\mathbf{S}_w = \sum_{j=1}^{c}\sum_{i=1}^{L}(\mathbf{y}_i - \boldsymbol{\mu}_j)(\mathbf{y}_i - \boldsymbol{\mu}_j)^T$. $\mathbf{y}_i$ is the ith sample of class j, $\boldsymbol{\mu}_j$ is the mean of class j, c is the number of classes, and L is the number of samples in each class. The between-class scatter matrix $\mathbf{S}_b$ is defined as $\mathbf{S}_b = \frac{1}{c}\sum_{j=1}^{c}(\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^T$, where $\boldsymbol{\mu}$ represents the means of all classes. With the obtained enhancer $\mathbf{M}_{lda}$, a $(c-1)$-dimensional vector $\mathbf{z}$ can be obtained such that

$$\begin{aligned}\mathbf{z} &= \mathbf{M}_{lda}^T\mathbf{y}^d = \mathbf{M}_{lda}^T[(\mathbf{M}_{pca}^d)^T\mathbf{x}] \\ &= (\mathbf{M}_{pca}^d\mathbf{M}_{lda})^T\mathbf{x} = \mathbf{M}_e^T\mathbf{x},\end{aligned} \tag{16}$$

where $\mathbf{z}$ is another compact version of the noise residual $\mathbf{x}$; $\mathbf{M}_e = \mathbf{M}_{pca}^d\mathbf{M}_{lda}$ is the refined SPN extractor which is used for extracting $\mathbf{z}$ directly from the original $\mathbf{x}$. In most cases, $c-1$ would be much smaller than d so that $\mathbf{z}$ would be more compact than $\mathbf{y}^d$.

4. Source camera identification

The camera identification process using the proposed compact features are summarized in Algorithms 1 and 2. We call the feature vectors $\mathbf{y}^d$ and $\mathbf{z}$ produced by Algorithms 1 and 2 as "PCA-SPN" and "LDA-SPN", respectively, throughout the rest of this paper. As mentioned earlier, the complexity of calculating correlation is proportional to the feature size. Considering that the size of PCA-SPN ($\mathbf{y}^d \in \mathbb{R}^d$) and LDA-SPN ($\mathbf{z} \in \mathbb{R}^{c-1}$) are both much lower than that of the original noise residual ($\mathbf{x} \in \mathbb{R}^m$), using either $\mathbf{y}^d$ or $\mathbf{z}$ in place of the original $\mathbf{x}$ would lead to approximately a m/d or $m/(c-1)$ times gain in speed in the matching phase.

In addition, given a required false positive rate, the detection thresholds $\tau_{\mathbf{y}}$ and $\tau_{\mathbf{z}}$ for the PCA-SPN ($\mathbf{y}^d$) and LDA-SPN ($\mathbf{z}$) can be determined by using the Neymann–Pearson criterion approach [31].

5. Experiments

In this section, we carry out experiments on the Dresden image database [40] to validate the feasibility of the proposed methods. First we evaluate and discuss some main parameters, which play key roles in the proposed methods. Significant performance gain is achieved by using the proposed training construction process, which can suppress the unwanted noise. After that we plot

Algorithm 1 SPN Feature Extraction through PCA

Symbols:
m: The original dimensionality of the SPN;
L: The number of training samples per camera;
c: The number of cameras;
n: The number of total training samples ($n = L \times c$).

1. Perform the training set construction procedure (refer to Section 3.1) to generate a set of training samples $\{\{\mathbf{x}_{ij}\}_{i=1}^{L}\}_{j=1}^{c} \in \mathbb{R}^m$.
2. Use the fast method mentioned in Section 3.2 to estimate the eigenvectors $\{\mathbf{v}_k\}_{k=1}^{n}$ and the eigenvalues $\{\lambda_k\}_{k=1}^{n}$.
3. Use the d eigenvectors corresponding to the d largest eigenvalues to build the SPN feature extractor $M_{pca}^{d} = [\mathbf{v}_1, ..., \mathbf{v}_d] \in \mathbb{R}^{m\times d}$.
4. Extract PCA-SPNs from all the training samples $\{\{\mathbf{x}_{ij}\}_{i=1}^{L}\}_{j=1}^{c}$ and the query noise residual $\mathbf{x}_q$ using $\mathbf{y}_{ij}^{d} = (\mathbf{M}_{pca}^{d})^T\mathbf{x}_{ij}$, $\mathbf{y}_{q}^{d} = (\mathbf{M}_{pca}^{d})^T\mathbf{x}_{q}$.
5. Estimate the reference PCA-SPN for camera C_j using $\mathbf{y}_{C_j}^{d} = \frac{1}{L}\sum_{i=1}^{L}\mathbf{y}_{ij}^{d}$.
6. Calculate the NCC value $\rho(\mathbf{y}_q^d, \mathbf{y}_{C_j}^d)$ between the query $\mathbf{y}_q^d$ and each reference $\mathbf{y}_{C_j}^d$ using Eq.(4).
7. Accept H_0 if $\rho(\mathbf{y}_q^d, \mathbf{y}_{C_j}^d) < \tau_{\mathbf{y}}$, otherwise accept H_1 (H_0 and H_1 are described in Section 2.3).

Algorithm 2 SPN Feature Enhancement through LDA

1.-4. Same as Step 1-4 of Algorithm 1.

5. Use the PCA-SPNs $\{\{\mathbf{y}_{ij}^{d}\}_{i=1}^{L}\}_{j=1}^{c}$ to estimate the transformation matrix $\mathbf{M}_{lda}$ using Eq. (15).
6. Extract LDA-SPNs from all the training samples $\{\{\mathbf{x}_{ij}\}_{i=1}^{L}\}_{j=1}^{c}$ and the query $\mathbf{x}_q$ using $\mathbf{z}_{ij} = \mathbf{M}_e^T\mathbf{x}_{ij}$, $\mathbf{z}_q = \mathbf{M}_e^T\mathbf{x}_q$.
7. Calculate the NCC value $\rho(\mathbf{z}_q, \mathbf{z}_{C_j})$ between query $\mathbf{z}_q$ and each reference $\mathbf{z}_{C_j}$ using Eq. (4).
8. Accept H_0 if $\rho(\mathbf{z}_q, \mathbf{z}_{C_j}) < \tau_{\mathbf{z}}$, otherwise accept H_1.

the histogram of intra-class and inter-class correlations to demonstrate the effectiveness of the PCA/LDA features. Based on several popular SPN algorithms, we also use our methods as a post-processing framework, and we also compare the dimensionality of different features under the same situation so as to evaluate the compactness of different types of features. Finally, the performance in terms of computational efficiency of the proposed methods are reported.

5.1. Experimental setup

In this work, images taken by 36 cameras from the Dresden image database [40] are used. As listed in Table 1, we can see these 36 cameras are of 12 different models, each having 1–5 devices.

A total of 7200 images from these 36 cameras are involved in our experiments. Each camera contributes 200 images, including 150 images with varying scene details (i.e., textured images) and 50 flat-field images. We consider two scenarios with different types of reference images (i.e., flat-field and textured), as shown in Table 2. For each image, a blocks of 512×512 pixels cropped from the center is used in the experiments so as to avoid the vignetting effect [33].

For each image block, we extract the noise residuals from three color channels (i.e., red, green and blue channel) and combine them by using the following linear combination to form a grayscale version, such that

$$\mathbf{x} = 0.299\mathbf{x}_R + 0.587\mathbf{x}_G + 0.114\mathbf{x}_B, \quad (17)$$

where $\mathbf{x}_R$, $\mathbf{x}_G$ and $\mathbf{x}_B$ are the noise residuals extracted from the red, green and blue channel, respectively.

In our experiments, the noise residuals extracted with the methods in [3] (Basic), [4] (MLE), [24] (BM3D) and [26] (PCAI8) are served as the original SPNs. SEA [30] is applied to enhance the reference SPNs and the training samples for PCA-SPN and LDA-SPN. The results are compared against the SPN Digest of [14].

NCC defined in Eq. (4) is used to measure the similarity in the SCI tasks.

Table 1
36 cameras involved in our experiments.

Camera models	Number of devices	Resolution
Canon_Ixus55	1	2592 × 1944
Canon_Ixus70	3	3072 × 2304
Olympus_mju_1050SW	5	3648 × 2736
Pentax_OptioA40	4	4000 × 3000
Pentax_OptioW60	1	3648 × 2736
Praktica_DCZ5.9	5	2560 × 1920
Rollei_RCP_7325XS	3	3072 × 2304
Samsung_L74wide	3	3072 × 2304
Samsung_NV15	3	3648 × 2736
Sony_DSC_H50	2	3456 × 2592
Sony_DSC_T77	4	3648 × 2736
Sony_DSC_W170	2	3648 × 2736

Table 2
The setup of two SCI scenarios.

Scenario	Reference images per camera	Query images per camera
1	50 flat-field images	100 textured images
2	50 textured images	100 textured images

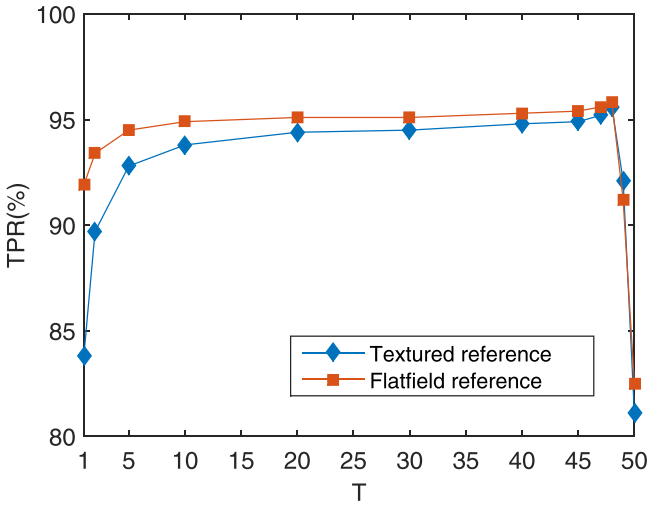


Fig. 2. The TPR (with the FPR fixed at 10^{-3}) of the PCA-SPN obtained from BM3D w.r.t. different setting of parameter T and and different reference types.

5.2. Parameter settings and discussions

In this work, one of the most important parameters is the number of noise residuals (T in Eq. (6)) used to estimate a training sample (also referred to as the random subset size). As discussed in Section 3.1, we set T to a relatively large number (i.e., $T \to N$, and $N = 50$ in this paper) so as to minimize the impact of scene details and random noise. Fig. 2 depicts the performance sensitivity (i.e., True Positive Rate (TPR) with the False Positive Rate (FPR) fixed at 10^{-3}) to T in the two SCI scenarios described in

Table 3
The dimensionality d of PCA-SPNs obtained from different SPN methods w.r.t. different setting of T and different reference types.

Method	Flat-field		textured	
	$T = 20$	$T = 48$	$T = 20$	$T = 48$
Basic	1042	609	1159	867
MLE	1013	605	1138	863
BM3D	1029	598	1148	848
PCAI8	1066	663	1148	860

Table 2. We can see that generally the performance based on PCA-SPN from BM3D features is not very sensitive to the setting of T (i.e., its performance is stable in a wide range of value of T [20, 48]). It improves slightly with an increasing value of T, reaching the peaks for both scenarios (i.e., with flat-field/textured reference) when $T = 48$. It is worth noting the result with $T = 1$ is the case without applying the proposed training set construction process, and the large performance gap (e.g., when compared with $T = 48$) indicates the effectiveness of our proposed training set construction process, especially for textured references. It is also interesting to see that the TPR drops dramatically when $T >= 49$, since when $T \rightarrow N$ ($N = 50$), all the training samples from the same camera become similar. Especially, when $T = N$ all the training samples from the same camera would become exactly the same. In this case, we literally have only one training sample per camera, and the training set is not large enough to learn the effective feature representation [41]. Therefore, to minimize SPN distortion, we set T to 48 throughout the rest of the paper.

It is also interesting to discuss d, the dimensionality of PCA-SPN in different cases. Clearly, we prefer d to be as small as possible without compromising the identification accuracy. d is determined by two main factors, namely the percentage of the total variance retained in Eq. (12) and the quality of the training set. As shown in Eq. (12), the value of d is affected by the percentage of the total variance that we aims to preserve (i.e., 98% in this paper). The less percentage that is retained, the smaller value of d would be. Table 3 shows the dimensionality d of the PCA-SPNs obtained from different SPN extraction methods with respect to different settings of T (i.e., $T = 20$ and $T = 48$), for two types of reference images. In both flat-field and textured training sets, we can see that the dimensionality d of PCA-SPNs decrease when T is larger. One reason is that, with a larger T, according to Eq. (8) we can see that the quality of the training set tend to be better (i.e., lower $\sigma^2_{\mathbf{x}'}$), thus the energy of the true SPN is more concentrated in the transformed domain. As a result, the SPN feature extractor requires less leading eigenvectors to cover the 98% of the total energy. Similarly, flat-field reference images (with training samples of higher quality) also tends to have a more compact representation than their textured counterpart, as shown in Table 3. It is worth mentioning that d is insensitive to the size of original SPN. According to our experimental results, the PCA-SPN derived from large image blocks has a similar size to the one from small image blocks. This observation indicates that the PCA-SPN is compressed more effectively when its original SPN is extracted from larger image blocks.

5.3. Distributions of intra-class and inter-class correlations

We evaluate the effectiveness of different features in terms of the distribution of their inter/intra-class correlations. A great separation between intra-class and inter-class distributions of a feature suggests the feature's high discriminative power. Experiments are conducted using 3 different types of SPNs (i.e., original SPN, PCA-SPN, and LDA-SPN) in the 2 SCI scenarios (with flat-field/textured reference as listed in Table 2). Results are reported in Fig. 3, from which we can see the means of the intra-class correlations are significantly increased by using PCA-SPN and LDA-SPN, when compared with the results based on the original SPNs. Specifically, for the two SCI scenarios, the application of PCA increases the mean of the intra-class correlations from 0.046 to 0.564 for the flat-field references while from 0.033 to 0.412 when only given the textured images as reference. The means of the intra-class correlations can be further boosted by using LDA-SPN owing to its supervised-learning nature, to 0.883 and 0.838, respectively, in the two scenarios.

The increase in the mean of the intra-class correlations results in the rightward shift of the intra-class distribution, which widens the separation between the intra/inter-class similarity distributions. However, the variance of the inter-class correlations is also increased in the case of using PCA-SPNs and LDA-SPNs. For example, in the case with flat-field references, the inter-class variance for PCA-SPN and LDA-SPN are 7.8×10^{-4} and 6.8×10^{-3}, respectively, which are higher than that of the original SPNs, 5.4×10^{-6}. However, the increase in variance are trivial when compared to the displacements of the means of the intra-class correlations (i.e., $0.564 - 0.046 = 0.518$ and $0.883 - 0.046 = 0.837$) away from the inter-class mean. This suggests the benefits of applying PCA-SPN and LDA-SPN in the SCI tasks. This is clearly reconfirmed in Fig. 3, where the overlapping area between the intra-class and inter-class distributions of PCA-SPN and LDA-SPN are much smaller, making the two distributions more separated (especially with LDA-SPN).

In addition, when using the original SPN (as shown in the first columns of Fig. 3), the intra-class distribution has small peaks in the overlapping area, which is mainly due to the small negative correlation exhibited among the matching SPN pairs. These small correlations are probably caused by the strong distortions due to scene details in some query images. Nevertheless, when using PCA-SPN and LDA-SPN (as shown in the figures from the last two columns), the numbers of small negative intra-class correlations are significantly reduced. As a result, the overlapping area decreases substantially, again reconfirming the merit of PCA-SPN and LDA-SPN. Moreover, since the separation is mainly caused by the rightward shift of the intra-class distribution, which has a major influence on the False Rejection Rate (FRR). As such, PCA-SPNs and LDA-SPNs have particular advantage in the situations where low FRR is preferred.

5.4. Performance comparison – accuracy

We can use the afore-mention methods (i.e., training set construction and PCA/LDA-based SPN feature extraction) as a post-processing for the existing SPN extraction methods. For evaluation purpose, here we report the performance (in terms of ROC curves) of four popular methods, namely, Basic [3], MLE [5], BM3D [25] and PCAI8 [26] combined with and without the proposed post-processing method. Moreover, since our method aims to compress the size of SPNs, we also present another SPN compression method (i.e., SPN Digest [14]) for comparison. SPN digest is primarily formed by retaining the top k largest elements from a m-dimensional SPN ($k < m$). Therefore, the size of SPN digest is k, which is lower than that of the original SPN. While a reference SPN Digest not only contains top k elements but also the corresponding locations of these k elements. This location information is used to extract the digests from the query SPNs so that it ensures the reference and query digest are extracted from the same locations. In this experiment, we set k/m equal to 10% and 20%.

In this experiment, the overall ROC curve is used for performance comparison. The number of true positive decisions and false positive decisions are first recorded for each camera. A true positive decision is made when hypothesis H_1 is true and H_1 is accepted, while a false positive decision is made when H_0 is true but H_1 is accepted. The total number of true decisions and false de-

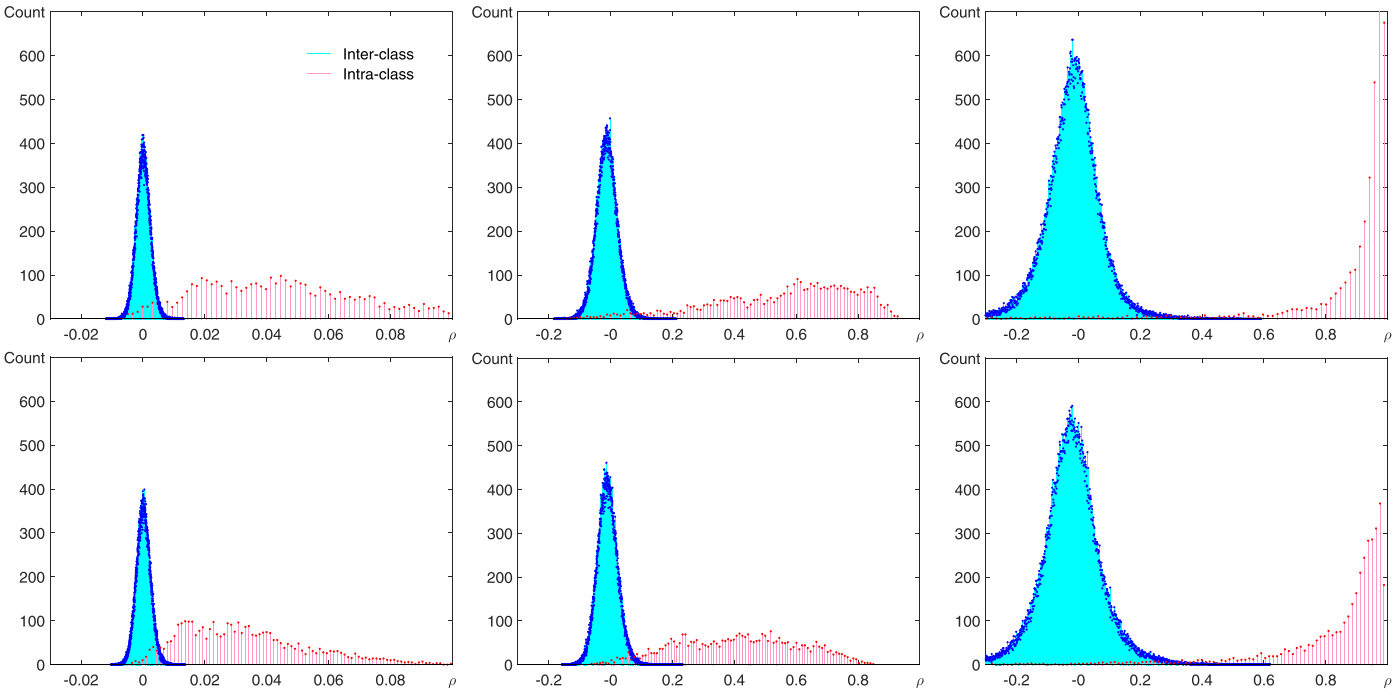


Fig. 3. Distributions of the inter/intra-class correlations w.r.t. different features (i.e., original SPN, PCA-SPN and LDA-SPN from left to right) and different reference types (1st row: flat-field reference and 2nd row: textured reference).

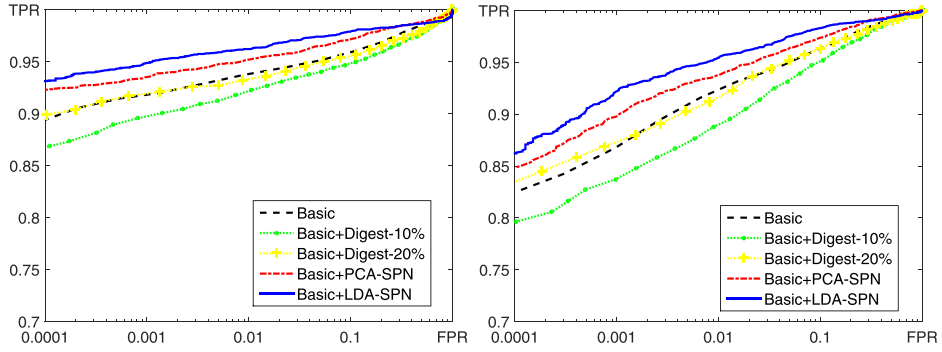


Fig. 4. Overall ROC curves of different types of features based on the Basic SPN extraction (Left: flat-field reference; Right: textured reference). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

cisions are calculated and then used to calculate the true positive rate $\mathcal{P}_{tp}$ and false positive rate $\mathcal{P}_{fp}$, respectively. Since the same number of images by each camera are use in our experiment, we can simply calculate the $\mathcal{P}_{tp}$ and $\mathcal{P}_{fp}$ for a threshold as follows

$$\mathcal{P}_{tp} = \frac{\sum_{i=1}^{c} \mathcal{D}_{tp}^{i}}{\mathcal{T}}, \mathcal{P}_{fp} = \frac{\sum_{i=1}^{c} \mathcal{D}_{fp}^{i}}{(c-1)\mathcal{T}}, i = 1, 2, \ldots, c, \tag{18}$$

where c is the number of cameras; $\mathcal{T}$ is the number of query images from all cameras; $\mathcal{D}_{tp}^{i}$ and $\mathcal{D}_{fp}^{i}$ are the number of true positive decisions and false positive decisions made for camera C_i. By varying the detection threshold from the minimum to maximum value, we can obtain the overall ROC curve. In real-world forensic applications, it is often necessary to ensure a sufficiently low FPR. Therefore, we plot the horizontal axis of the overall ROC curve in the logarithmic scale.

Fig. 4 shows the overall ROC curves of different features based on the Basic SPN extraction method [3] in the two SCI scenarios described in Table 2, i.e., with flat-field/textured reference images. The black, green, yellow, red and blue curve indicates the performance of the original SPN (i.e., Basic), SPN Digest-10%, SPN Digest-20%, PCA-SPN and LDA-SPN, respectively. In both SCI scenarios, we can see that SPN Digest performances very closely to the original SPN when 20% of the top largest elements are retained, but its performance degrades rapidly when the amount of the retained elements is reduced to 10%. On the other hand, the LDA-SPN (blue line) achieves the best ROC performance regardless of the type of reference images, while the PCA-SPN (red line) takes the second place. The same observation can be made from Figs. 5 to7 when different SPN extraction methods (i.e., MLE, BM3D and PCIA) are used respectively.

5.5. Performance comparison – compactness

In this section, we compare the compactness of different types of features. The dimensionality of SPN Digest is determined by k/m. Therefore, in the case of using the image block of 512×512 pixels, the size of SPN Digest-10% and -20% are 26,215 and 52,429, respectively. As listed in Table 3, with flat-field references, the dimensionality of PCA-SPNs based on Basic, MLE, BM3D and PCAI are 609, 605, 598 and 663, respectively, and they are accordingly increased to 867, 863, 848 and 860 when the references are textured images. The size of LDA-SPN is always equal to $c-1$, which is 35

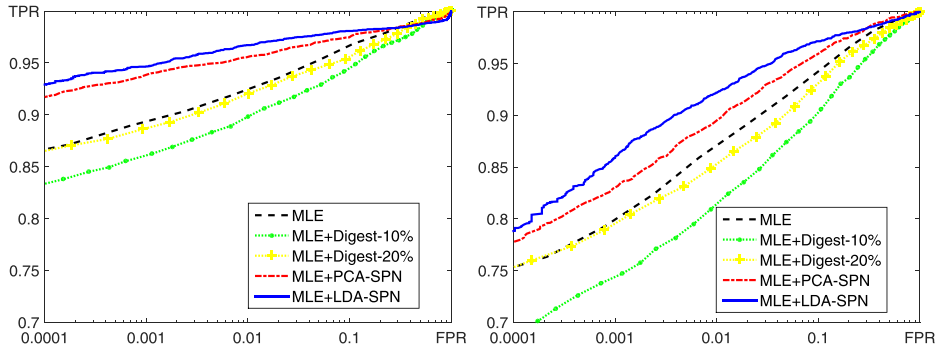


Fig. 5. Overall ROC curves of different types of features based on the MLE SPN extraction (Left: flat-field reference; Right: textured reference). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

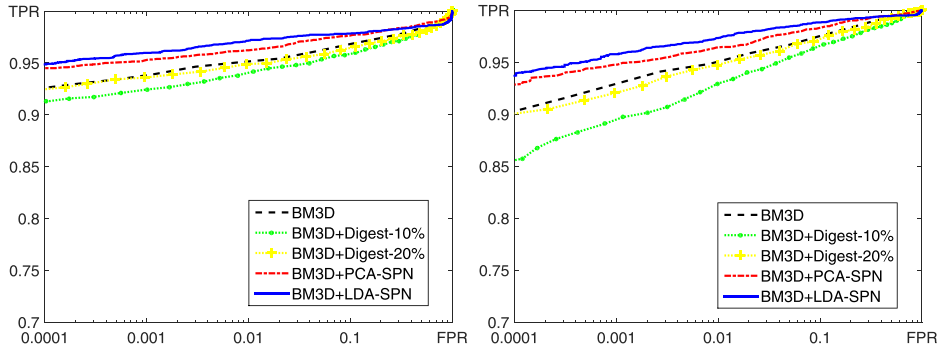


Fig. 6. Overall ROC curves of different types of features based on the BM3D SPN extraction (Left: flat-field reference; Right: textured reference). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

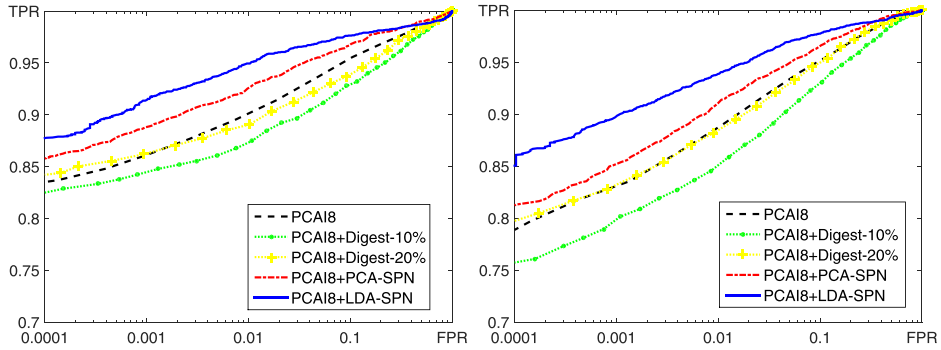


Fig. 7. Overall ROC curves of different types of features based on the PCAI SPN extraction (Left: flat-field reference; Right: textured reference). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

in this experiment. This observation shows that the dimensionality of SPN Digest is much higher than that of PCA-SPN and LDA-SPN. Considering the results obtained in Section 5.4, we can conclude that both PCA-SPN and LDA-SPN are superior to SPN Digest in terms of compactness and identification accuracy. Experimental results also validate that the proposed SPN feature extraction method can be used as a general post-processing method applied after various SPN extraction methods in the SCI task.

Table 4
Computational cost (seconds) of SPN Digest-20% and different types of features produced by BM3D.

Features	I/O operations	Feature extraction	Matching	Total
Original SPN	0.08	0	12425.48	12425.56
SPN Digest	0.03	1236.25	1770.12	3006.40
PCA-SPN	1.52	4459.35	238.35	4699.22
LDA-SPN	0.08	399.71	154.48	554.27

5.6. Performance comparison – computational complexity

An efficient SCI system plays an important role when (i) there is a database with a large number of references and (ii) thousands of query SPNs are required to be identified. In order to testify the proposed framework on a large database, we perform this experiment on a synthetic database, which contains 180 cameras derived from the 36 cameras in Table 1 based on the fact that SPNs are location dependent (i.e., SPN blocks cropped from different locations of the same full-sized SPN are not correlated). To build this synthetic database, we first estimate the full-sized reference SPN for each camera in Table 1. Then we crop 5 SPN blocks of 512×512 pixels from different locations of each full-sized reference SPN and treat them as references for 5 different cameras, so that eventually we obtain 180 reference SPNs in total. We also generate 18,000 query SPNs in the same manner.

Table 5
The size (MB) of data required to be loaded for SPN Digest-20% and different types of features produced by BM3D.

Features	Data size	
	Feature extractor	References vectors
Original SPN	0	45.05
SPN Digest	9.01	9.01
PCA-SPN	621.21	0.43
LDA-SPN	44.80	0.03

Table 4 shows the running time for matching the 18,000 query samples with the simulated 180 cameras w.r.t. different types of features. In this case, the size of the original SPN, SPN Digest, PCA-SPN and LDA-SPN are $m = 262,144$, $k = 52,429$, $d = 2,484$ and $c - 1 = 179$, respectively. This experiment is conducted on the same PC with an Intel Core i5 3.20 GHz processor and 16G RAM. In order to reduce the storage requirement, all the data in this experiment is stored in the uint8 data type. To do so, we first project all the data onto the range of [0,255] and then convert the data type from double-precision floating-point to uint8 before storing.

To quantify the efficiency of an identification system, three factors are considered in this experiment. The first factor is "I/O operations", which includes the cost of loading the references and the SPN feature extractors into memory for processing. The second one is "Feature Extraction", indicating the time spend on producing SPN Digest, PCA-SPNs or LDA-SPNs from the 18,000 query noise residuals. The third factor is the computational cost for calculating the similarity between the 18,000 query samples and the 180 references, which is referred to as "Matching". The overall computational cost is presented as "Total".

As shown in Table 4, PCA-SPN incurs the highest computational cost in terms of I/O operations. It is because the data needs to be loaded into the memory. It includes not only the 180 m-dimensional reference vectors but also an $m \times d$-dimensional feature extractor ($\mathbf{M}_{pca}^{d}$).

As shown in Table 5, PCA-SPN needs a very small space to store its 180 reference vectors (0.43MB) but a relatively huge space for the feature extractor (621.21MB). With such a large amount of data in total, it is not surprising to see PCA-SPN incurs the highest computational cost in terms of I/O operations. LDA-SPN also needs to load a feature extractor ($\mathbf{M}_{lda}$), but its size is only $m \times (c-1)$ so that the space it occupies is much smaller than that of PCA-SPN, which is 44.80MB. Moreover, since the size of LDA-SPN is only $c - 1$, its storage overhead for the 180 reference vectors (0.03MB) is the lightest among all the features. In this experimental setting, the total storage requirement of LDA-SPN (44.83MB) is just slightly lower than that of the original SPN (45.05MB), but this margin will grow in a linear manner w.r.t. the increasing number of cameras.

SPN Digest requires the smallest storage among these 4 types of features. As mentioned earlier, the digest of a normal-sized reference SPN consists of not only the k top largest elements but also the corresponding location information of these k elements. This location information will be used to extract query digests from the query SPNs so that the location information of each reference digest can be also treated as a feature extractor. Therefore, when using SPN Digest, the data to be loaded includes not only 180 k-dimensional reference digests but also 180 corresponding k-dimensional SPN feature extractors, which take up a space of 18.02 MB in total. As a result, SPN Digest incurs the lowest computational cost in I/O operations (as shown in Table 4).

As mentioned in [42], the process of matching a query feature with all the references in the database has complexity proportional to the product of the number of references and the feature size. For example, when using the Original SPN, the complexity of the matching phase would be $O(\mathrm{cm})$. Since the number of query samples and references in the database are fixed in this case, therefore LDA-SPN, which is of the lowest dimensionality, incurs the least computational cost in the matching phase. PCA-SPN takes the second place, followed by SPN Digest and Original SPN. Although LDA-SPN, PCA-SPN and SPN Digest incur an extra computational cost in the feature extraction process, but with all aspects taken into account, we can see from Table 4 that replacing Original SPN with LDA-SPN, PCA-SPN or SPN Digest can significantly reduce the overall computational cost.

Bear in mind, these above-mentioned post-processing methods would also incur an extra computational cost in the training process or the process of estimating the optimal SPN Digest. However, compared to the processes that have to be conducted on-line (i.e., the processes listed in Table 4), PCA/LDA training or SPN Digest estimation can be performed off-line, and there is no need to re-run these processes as long as the population of database does not change. Moreover, the efficiency of the off-line operations of an SCI system is generally less important when it is compared to the identification accuracy or the on-line matching efficiency. Therefore, the computational cost of the off-line operations, i.e., PCA/LDA training and SPN Digest estimation, are not counted in this experiment.

6. Conclusion

In this paper, we introduced and evaluated the concept of PCA de-noising in the SCI task. Based on this concept, an effective framework for de-noising and compressing full-sized SPNs is proposed. We also proposed a training set construction method that minimizes the impact of interfering artifacts, which plays an important role in learning the SPN feature extractor that is insensitive to various unwanted noise. Both theoretical derivations and experimental results suggest that our methods can be used as a general post-processing framework for effective and efficient source camera identification. It is worth mentioning that the proposed framework also achieves very competitive performance in the challenging tasks when only textured references are available, which is usually the case in real-world applications. However, so far we focus on the case that the reference SPNs of all the cameras in question are included in the training set, while in real-world forensic applications, reference SPNs of new cameras will continuously be added to the database. In this case, the proposed system needs to re-perform the training process with the new cameras or reference SPNs of the cameras involved so as to maintain the identification accuracy. A new line for future research is to develop a new methodology that can progressively update the previously trained SPN feature extractor to accommodate the newly received reference SPNs without having to re-train the entire expanded set.

Acknowledgments

This work is supported by the EU Horizon 2020 – Marie Sklodowska-Curie Actions through the project entitled Computer Vision Enabled Multimedia Forensics and People Identification (Project No. 690907, Acronym: IDENTITY).

References

[1] M.J. Sorrell, Digital camera source identification through JPEG quantisation, in: C.-T. Li (Ed.), Multimedia Forensics and Security, Hershey, PA: IGI Global, 2009.
[2] P. Blythe, J. Fridrich, Secure digital camera, in: Proceedings of the 2004 Digital Forensic Research Workshop, 2004, pp. 11–13.
[3] J. Lukas, J. Fridrich, M. Goljan, Digital camera identification from sensor pattern noise, IEEE Trans. Inf. Forensics Secur. 1 (2) (2006) 205–214.
[4] M. Chen, J. Fridrich, M. Goljan, J. Lukas, Determining image origin and integrity using sensor noise, IEEE Trans. Inf. Forensics Secur. 3 (1) (2008) 74–90.
[5] C.-T. Li, Source camera identification using enhanced sensor pattern noise, IEEE Trans. Inf. Forensics Secur. 5 (2) (2010) 280–287.

[6] X. Kang, Y. Li, Z. Qu, J. Huang, Enhancing source camera identification performance with a camera reference phase sensor pattern noise, IEEE Trans. Inf. Forensics Secur. 7 (2) (2012) 393–402.
[7] A.J. Cooper, Improved photo response non-uniformity (PRNU) based source camera identification, Forensic Sci. Int. 226 (1) (2013) 132–141.
[8] L.-H. Chan, N.-F. Law, W.-C. Siu, A confidence map and pixel-based weighted correlation for PRNU-based camera identification, Digit. Invest. 10 (3) (2013) 215–225.
[9] L.T. Van, S. Emmanuel, M. Kankanhall, Identifying source cell phone using chromatic aberration, in: Proceedings of the 2007 IEEE International Conference on Multimedia and Expo, 2007, pp. 883–886.
[10] A. Swaminathan, M. Wu, K.R. Liu, Nonintrusive component forensics of visual sensors using output images, IEEE Trans. Inf. Forensics Secur. 2 (1) (2007) 91–106.
[11] O. Celiktutan, B. Sankur, I. Avcibas, Blind identification of source cell-phone model, IEEE Trans. Inf. Forensics Secur. 3 (3) (2008) 553–566.
[12] S. Bayram, H. Sencar, N. Memon, Sensor fingerprint identification through composite fingerprints and group testing, IEEE Trans. Inf. Forensics Secur. 10 (3) (Mar. 2015) 597–612.
[13] Y. Hu, C.-T. Li, Z. Lai, Fast source camera identification using matching signs between query and reference fingerprints, Multimed. Tools Appl. 74 (18) (2015) 7405–7428.
[14] M. Goljan, J. Fridrich, Sensor fingerprint digests for fast camera identification from geometrically distorted images, in: Proceedings of the 2013 IS&T/SPIE Electronic Imaging, 2013 86650B.1-86650B.10.
[15] S. Bayram, H. Sencar, N. Memon, Efficient sensor fingerprint matching through fingerprint binarization, IEEE Trans. Inf. Forensics Secur. 7 (4) (2012) 1404–1413.
[16] D. Valsesia, G. Coluccia, T. Bianchi, E. Magli, Compressed fingerprint matching and camera identification via random projections, IEEE Trans. Inf. Forensics Secur. 10 (1) (2015) 1472–1485.
[17] D. Valsesia, G. Coluccia, T. Bianchi, E. Magli, Large-scale image retrieval based on compressed camera identification, IEEE Trans. Multimed. 17 (1) (2015) 1439–1449.
[18] S. Taspinar, H.T. Sencar, S. Bayram, N. Memon, Fast camera fingerprint matching in very large databases, in: Proceedings of the 2017 IEEE International Conference on Image Processing, Beijing, China, 17–20, 2017.
[19] R. Li, C.-T. Li, Y. Guan, A compact representation of sensor fingerprint for camera identification and fingerprint matching, in: Proceedings of the 2015 IEEE International Conference o. Acoustics, Speech, and Signal Processing, Brisbane, Australia, 19–24, 2015, pp. 1777–1781.
[20] R. Li, C.-T. Li, Y. Guan, Incremental update of feature extractor for camera identification, in: Proceedings of the 2015 IEEE International Conference on Image Process., Quebec City, Canada, 27–30, 2015.
[21] D. Zhang, R. Lukac, X. Wu, D. Zhang, PCA-based spatially adaptive denoising of CFA images for single-sensor digital cameras, IEEE Trans. Image Process. 18 (4) (2009) 797–812.
[22] L. Zhang, W. Dong, D. Zhang, G. Shi, Two-stage image denoising by principal component analysis with local pixel grouping, Pattern Recognit. 43 (4) (2010) 1531–1549.
[23] M. Mihcak, I. Kozintsev, K. Ramchandran, Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising, in: Proceedings of the 1999 IEEE International Conference o Acoustics, Speech, and Signal Processing, 6, March, 1999, pp. 3253–3256.
[24] G. Chierchia, S. Parrilli, G. Poggi, C. Sansone, L. Verdoliva, On the influence of denoising in PRNU based forgery detection, in: Proceedings of the 2010 ACM Workshop on Multimedia in Forensics, Security and Intelligence, 7880, 2010, pp. 117–122.
[25] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3D transform-domain collaborative filtering, IEEE Trans. Image Process. 16 (8) (2007) 2080–2095.
[26] X. Kang, J. Chen, K. Lin, A. Peng, A context-adaptive SPN predictor for trustworthy source camera identification, EURASIP J. Image Video Process. 2014 (1) (2014) 1–11.
[27] W. Liu, W. Zeng, L. Dong, Q. Yao, Efficient compression of encrypted grayscale images, IEEE Trans. Image Process. 19 (4) (2010) 1097–1102.
[28] C.-T. Li, Y. Li, Color-decoupled photo response non-uniformity for digital image forensics, IEEE Trans. Circuits Syst. Video Technol. 22 (2) (2012) 260–271.
[29] M. Chen, J. Fridrich, M. Goljan, J. Lukas, Source digital camcorder identification using sensor photo response non-uniformity, in: Proceedings of the 2007 SPIE Electronic Imaging, Security, Steganography, Watermarking of Multimedia Contents IX, 6505, 2007 1G-1H.
[30] X. Lin, C.-T. Li, Preprocessing reference sensor pattern noise via spectrum equalization, IEEE Trans. Inf. Forensics Secur. 11 (1) (2016) 126–140.
[31] J. Fridrich, M. Goljan, Derivation of ROCs for composite fingerprints and sequential trimming, Dept. Elect. Comput. Eng., Binghamton Univ., Binghamton, NY, USA, Tech. Rep. [Online]. Available: http://www.ws.binghamton.edu/fridrich/Research/rocs.pdf. Jan. 2010.
[32] M. Goljan, Digital Camera Identification from Images – estimating false acceptance probability, in: Proceedings of the 2009 International Workshop Digital–forensics and Watermarking, 2009, pp. 454–468.
[33] C.-T. Li, R. Satta, Empirical investigation into the correlation between vignetting effect and the quality of sensor pattern noise, IET Comput. Vis. 6 (6) (2012) 560–566.
[34] K. Fukunaga, Introduction to Statistical Pattern Recognition, second ed., Academic, New York, 1991.
[35] M. Benito, D. Peña, A fast approach for dimensionality reduction with image data, Pattern Recognit. 38 (12) (2005) 2400–2408.
[36] C. Yang, L. Wang, J. Feng, On feature extraction via kernels, IEEE Trans. Syst. Man Cybern. B Cybern. 38 (2) (2008) 553–557.
[37] Y. Wang, K.N. Plataniotis, An analysis of random projection for changeable and privacy-preserving biometric verification, IEEE Trans. Syst. Man Cybern. B Cybern. 40 (5) (2010) 1280–1293.
[38] J. Jauregui, Principal component analysis with linear algebra(2012).
[39] J. Yang, J.-Y. Yang, Why can LDA be performed in PCA transformed space? Pattern Recognit. 36 (2) (2003) 563–566.
[40] T. Gloe, R. Böhme, The dresden image database for benchmarking digital image forensics, J. Digit. Forensic Pract. 3 (2–4) (2010) 150–159.
[41] A.M. Martínez, A. Kak, PCA versus LDA, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2) (2001) 228–233.
[42] M. Goljan, J. Fridrich, T. Filler, Managing a large database of camera fingerprints, in: Proceedings of the 2010 SPIE, Electronic Imaging, Media Forensics and Security XII, 7541, San Jose, 17–21, 2010, pp. 754108.01–754108.12.

Ruizhe Li received the B.Sc. degree in software engineering from Sichuan University, Chengdu, China, in 2010, and the Ph.D. degree in computer science from the University of Warwick, Coventry, U.K., in 2016. His research interests include digital forensics, multimedia security, image processing, machine learning, and computer vision.

Chang-Tsun Li received the B.Eng. degree in electrical engineering from National Defence University (NDU), Taiwan, in 1987, the M.Sc. degree in computer science from U.S. Naval Postgraduate School, USA, in 1992, and the Ph.D. degree in computer science from the University of Warwick, UK, in 1998. He was an associate professor of the Department of Electrical Engineering at NDU during 1998–2002 and a visiting professor of the Department of Computer Science at U.S. Naval Postgraduate School in the second half of 2001. He was a professor of the Department of Computer Science at the University of Warwick, UK, until Dec 2016. He is currently a professor of the School of Computing and Mathematics, Charles Sturt University, Australia. His research interests include multimedia forensics and security, biometrics, data mining, machine learning, data analytics, computer vision, image processing, pattern recognition, bioinformatics, and content-based image retrieval. The outcomes of his multimedia forensics research have been translated into award-winning commercial products protected by a series of international patents and have been used by a number of police forces and courts of law around the world. He is currently Associate Editor of the EURASIP Journal of Image and Video Processing (JIVP) and Associate of Editor of IET Biometrics. He involved in the organisation of many international conferences and workshops and also served as member of the international program committees for several international conferences. He is also actively contributing keynote speeches and talks at various international events.

Yu Guan obtained his Ph.D. degree in Computer Science from the University of Warwick, 2015. He is now a lecturer in Open Lab, School of Computing, Newcastle University. His research interests are machine learning and its applications in the areas of behaviour analysis, computer vision, biometrics and forensics. He is currently Associate Editor of ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT) and a regular reviewer for several top-tier journals such as IEEE T-PAMI, IEEE T-CSVT, IEEE T-IFS, IEEE T-CYB, etc.