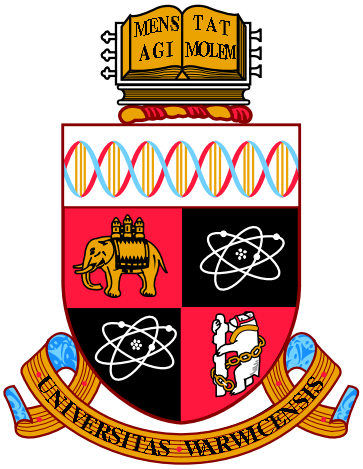


Learning based Forensic Techniques for Source Camera Identification

by

Ruizhe Li

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Computer Science

March 2016

THE UNIVERSITY OF WARWICK

Contents

Acknowledgments

First and foremost I would like to express my deepest sense of gratitude to my supervisor Prof. Chang-Tsun Li, who has been constantly supportive and inspiring during my PhD study. His great personality, unlimited patience and tolerance has educated me a lot more than scientific research.

I wish to express my sincere thankfulness to my annual progress panel members Dr. Victor Sanchez, Dr. Arshad Jhumka and Prof. Yongjian Hu for their guidance and valuable suggestions on my research.

I would also like to thank the colleagues at the University of Warwick, Dr. Yu guan, Dr. Xingjie Wei, Dr. Yi Yao, Mr. Ning Jia, Mr. Xin Guan, Mr. Alaa Khadidos, Mr. Xufeng Lin, Mr. Qiang Zhang, Mr. Roberto Leyvac, Mr. Shan Lin and Mr. Ching-Chun Chang for their kindness not just inside the lab.

I would like to express my deepest gratitude to my parents, Mr. Dongyang Li and Ms. Xin Ji for their love, understanding and encouragement throughout my life, which give me the strength to chase whatever I want. To my wife, Ms Lin Lin for being my side, and all the inspirations and love she brings to my life.

Last but not least, to my friends, Mr. Yile Liu, Ms. Simin Tian, Mr. Boyang Peng, Mr. Chengyu Yu, for making my graduate life more bearable and full of fun.

Declarations

I hereby declare that this dissertation entitled *Learning based forensic techniques for source camera identification* is an original work and has not been submitted for a degree or diploma or other qualification at any other University.

Abstract

In recent years, multimedia forensics has received rapidly growing attention. One challenging problem of multimedia forensics is source camera identification, the goal of which is to identify the source of a multimedia object, such as digital image and video. Sensor pattern noises, produced by imaging sensors, have been proved to be an effective way for source camera identification. Precisely speaking, the conventional SPN-based source camera identification has two application models: verification and identification. In the past decade, significant progress has been achieved in the tasks of SPN-based source camera verification and identification. However, there are still many cases requiring solutions beyond the capabilities of the current methods. In this thesis, we considered and addressed two commonly seen but less studied problems.

The first problem is the source camera verification with reference SPNs corrupted by scene details. The most significant limitation of using SPN for source camera identification is that SPN can be seriously contaminated by scene details. Most existing methods consider the contaminations from scene details only occur in query images but not in reference images. To address this issue, we propose a measurement based on the combination of local image entropy and brightness so as to evaluate the quality of SPN contained by different image blocks. Based on this measurement, a context adaptive reference SPN estimator is proposed to address the problem that reference images are contaminated by scene details.

The second problem that we considered relates to the high computational complexity of using SPN in source camera identification, which is caused by the high dimensionality of SPN. In order to improve identification efficiency without

degrading accuracy, we propose an effective feature extraction algorithm based on the concept of PCA denoising to extract a small set of components from the original noise residual, which tends to carry most of the information of the true SPN signal. To further improve the performance of this framework, two enhancement methods are introduced. The first enhancement method is proposed to take the advantage of the label information of the reference images so as to better separate different classes and further reduce the dimensionality. Secondly, we propose an extension based on Candid Covariance-free Incremental PCA to incrementally update the feature extractor according to the received images so that there is no need to re-conduct training every time when a new image is added to the database. Moreover, an ensemble method based on the random subspace method and majority voting is proposed in the context of source camera identification to tackle the performance degradation of PCA-based feature extraction method due to the corruption by unwanted interferences in the training set.

The proposed algorithms are evaluated on the challenging Dresden image database and experimental results confirmed their effectiveness.

Abbreviations

BM3D	Block-matching and 3D filtering
CAI	Context adaptive interpolation
CCD	Charge-coupled devices
CCIPCA	Candid covariance-free incremental PCA
CCN	Correlation over circular correlation norm
CFA	Color filter array
CLT	Central limit theorem
CMOS	Complementary-metal-oxide semiconductor
DCT	Discrete cosine transform
DFT	Discrete Fourier transform
EXIF	Exchangeable image file
FFT	Fast Fourier transform
FPN	Fixed pattern noise
FPR	False positive rate

JEET	Junction field-effect transistor
JL	Johnson-Lindenstrauss lemma
LDA	Linear discriminant analysis
MAP	Maximum A-Posteriori probability
MLE	Maximum-likelihood estimation
MvDA	Multi-view discriminant analysis
PCA	Principal component analysis
PCAFE	PCA-based feature extraction
PCAI	Predictor based on the context adaptive interpolation
PCE	Peak-to-correlation energy
PDF	Probability density function
PRNU	Photo-response non-uniformity noise
RSM	Random subspace method
ROC	Receiver operating characteristic
SCI	Source camera identification
SCV	Source camera verification
SPN	Sensor pattern noise
TPR	True positive rate
ULDA	Uncorrelated linear discriminant analysis
WGN	White Gaussian noise

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Nowadays, digital imaging devices, such as digital cameras, camcorders and cameras embedded in smart phones, are widely used in the modern world. In 2014, more than 1 billion imaging devices have been produced and sold. As a consequence, over one trillion digital images were taken in that year. With such enormous amount of digital images, the use of digital images as critical evidences in the fight against crime is increasing, making multimedia forensic investigations more frequent and important. Typically, multimedia forensics includes source device identification, source-oriented images classification, integrity verification or forgery detection, authentication, etc. Source camera identification is a very important branch of multimedia forensics, which aims to prove whether a given image or video was taken by a specific imaging device. This technique has been utilized to identify the origin of digital images or videos in many forensic cases, such as child pornography, movie piracy, proof of ownership, terrorist investigations, etc. For example, in an investigation of a child sexual

abuse case [1], a person was accused of taking a set of child pornography images, while this suspect refused to plea guilty and claimed that these child pornography images in his smart phones were downloaded from internet and not taken by him. The police applied a source camera identification technique and proved that these images were indeed taken by the suspect's smart phone. The suspect was finally convicted after a short trial.

1.2 Source Camera Identification

In order to trace the source of digital images or videos, many efforts have been made in the task of source camera identification. These proposed techniques can be roughly divided into three categories:

- **Metadata**. The simplest technique for identifying the source camera is to use the metadata embedded by the source camera [2]. For example, exchangeable image file (EXIF) [3] is a format for storing metadata in image and audio files. The EXIF header contains information of camera manufacturer, camera model and conditions under which the image was taken (exposure, date, time, etc.). The EXIF header is encoded in ASCII text, it can be directly read in the binary file or easily extracted by using many common photography tools, such as Adobe Photoshop and IrfanView. One can use this metadata to identify the model of the source camera (but not the specific camera). However, along with the wide use of these photography tools, header data can easily be changed or removed by users. In addition, metadata may not be available if the image is re-saved in a different format or re-compressed. For these reasons, metadata is not expected to be a reliable indicator of the image source.

- **Watermarking**. Another possible solution to the source camera identification problem is to use the digital watermark embedded in the image by the camera, which carries information about the source camera, the time when the image was taken, and even a biometric data of the person taking the image. There are a few camera manufacturers offering cameras with watermarking capabilities, which is called "Secure Camera", such as Epson PhotoPC 700/750Z, 800/800Z, 3000Z and Kodak DC-200, DC-260, DC-290 [4]. Such cameras transparently insert a digital watermark to each image or video they capture, thus one can determine whether a given image is taken by a specific secure camera by matching the watermark from the given image and the specific secure camera. This technique is very useful for proving ownership of a copyright work in the case with secure cameras, while it is of no help in tracking the source when images are taken by other cameras.
- **Manufacturer Specific Technique**. The third set of techniques relies on the intrinsic characteristics of digital cameras left in the image. Generally speaking, any traces left in the image by the processing components of the image acquisition pipeline have the potential to link the images to the source camera, such as sensor pattern noise (SPN) [5–14], lens aberration [15, 16], colour filter array (CFA) interpolation artifacts [17, 18], JPEG compression [2, 19], and the combination of several intrinsic characteristics [20, 21]. Among these traces, SPN has been proved as the most effective way for source camera identification, and has attracted the most attention in the research area. Compared with other intrinsic characteristics, SPN has several remarkable advantages:

1. **Uniqueness.** Every image or video taken by the same sensor exhibits the same SPN pattern, while any two imaging sensors would exhibit uncorrelated

SPN patterns even when they are from the same silicon wafer [6].

2. **Generality.** SPN is present in every digital image and video that captured by imaging sensors, regardless of the camera optics, settings, or the scene content [22].

3. **Universality.** All digital imaging sensors would exhibit SPN pattern, such as charge-coupled devices (CCD) sensor [23, 24], complementary-metal-oxide semiconductor (CMOS) sensor [25], junction field-effect transistor (JFET) sensor and CMOS-Foveon X3 sensor [6].

4. **Stability.** SPN is stable in time and under a wide range of physical conditions, such as ambient temperature or humidity [22].

5. **Robustness.** SPN can survive from a series of operations performed on the image such as blurring, scaling, compression, and even printing and scanning. Moreover, it is often beyond the capability of normal camera users to manipulate or remove this fingerprint from digital images or videos.

Due to these advantages, the SPN-based sensor fingerprint is quite popular in several digital forensic applications, such as source camera identification, image clustering, forgery detection, etc. In this thesis, our research interest is particularly focused on the area of the SPN-based source camera identification.

1.3 SPN-based Source Camera Verification and Identification

Precisely speaking, the conventional SPN-based source camera identification has two application models: verification and identification. Source Camera Verification (SCV) is the task that conducts one-to-one comparison to verify whether a given image or video is taken by a specific imaging device. It is especially useful in the court of law for addressing the question: *Is this image taken by the claimed camera?* In order to verify that a query image was taken by a camera, we first need to extract the SPN from the query image and estimate the reference SPN for the camera. Then the similarity between the query SPN and reference SPN is calculated. The decision is finally made by comparing the obtained similarity score with a predetermined threshold.

On the other hand, source camera identification (SCI) is the task that searches a database for an enrolled camera or SPN fingerprint to match the query image, i.e., one-to-many comparison. The goal of SCI is to answer the question: *Which camera in a database has taken the query image?* Different from verification, the identification system conducts multiple one-to-one comparisons between the query SPN and a lot of reference SPNs from the database, and returns a single match (the best match) as the most probable association with the query sample. Fig 1.1 shows the flowcharts of the SPN-based source camera verification and identification. As illustrated in Fig. 1.1, both the verification and identification system consist of three main stages: SPN extraction, reference SPN estimation and SPN matching, the details of which will be presented in Chapter 2.

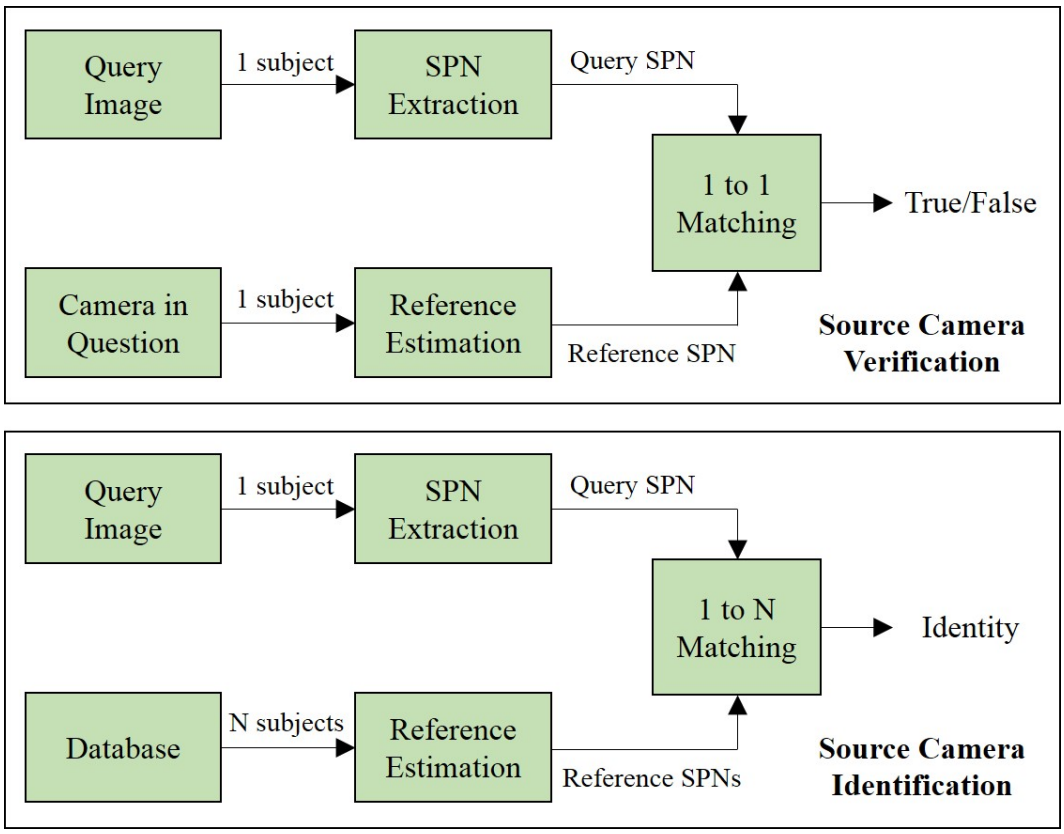


Figure 1.1: The flowcharts of source camera verification and identification system.

1.4 Challenges

The SPN-based source camera identification has been studied for more than a decade by the multimedia forensic community. Many existing SPN-based methods in the literature can accurately link digital images or videos to the specific cameras that acquired them. However, there are still many cases requiring solutions beyond the capabilities of the current methods. In this thesis, we select the following two existing problems as the research topic since they are both commonly seen and less studied.

1.4.1 Reference Images Corrupted by Scene Details

In real-world applications, SPN can be easily contaminated by many interferences, which would decrease the identification accuracy. Those interference factors can be roughly summarised into two categories. On the one hand, SPN can be affected by the characteristics produced in the imaging acquisition process, such as shot noise, read-out noise, quantization noise, CFA interpolation artifacts, etc [6]. Most middle-end and high-end cameras have the capability to suppress these contaminations, thus the impact of these contaminations is relatively low for the identification result, especially in the uncompressed high quality images. However, it is difficult to fully remove these contaminations.

On the other hand, SPNs can be contaminated by image content, i.e., scene edges and textures. As mentioned in [9], both SPN and scene textures appear as the high-frequency signal, thus in the SPN extraction stage not only the true SPN components but also the trace of scene textures would be extracted. The interference of scene textures can be very serious as its magnitude is usually far greater than that of the true SPN signal in the noise residual [9]. Moreover, it is very common in real-world environments as most normal images would contain a certain amount of scene textures. But this contamination can be easily avoided. For example, if we have access to the camera to be identified, we can use it to take some images of smooth scenes, such as blue sky and flat wall. By doing so, we can actively avoid the acquired images from contaminations of scene textures and consequently extract some clean SPNs. Examples of SPNs extracted from different kinds of images are shown in Fig. 1.2. Fig. 1.2 (a) and (b) are a blue sky image and an image with strong scene details taken by a Canon IXUS70, respectively. Fig. 1.2 (c) and (d)

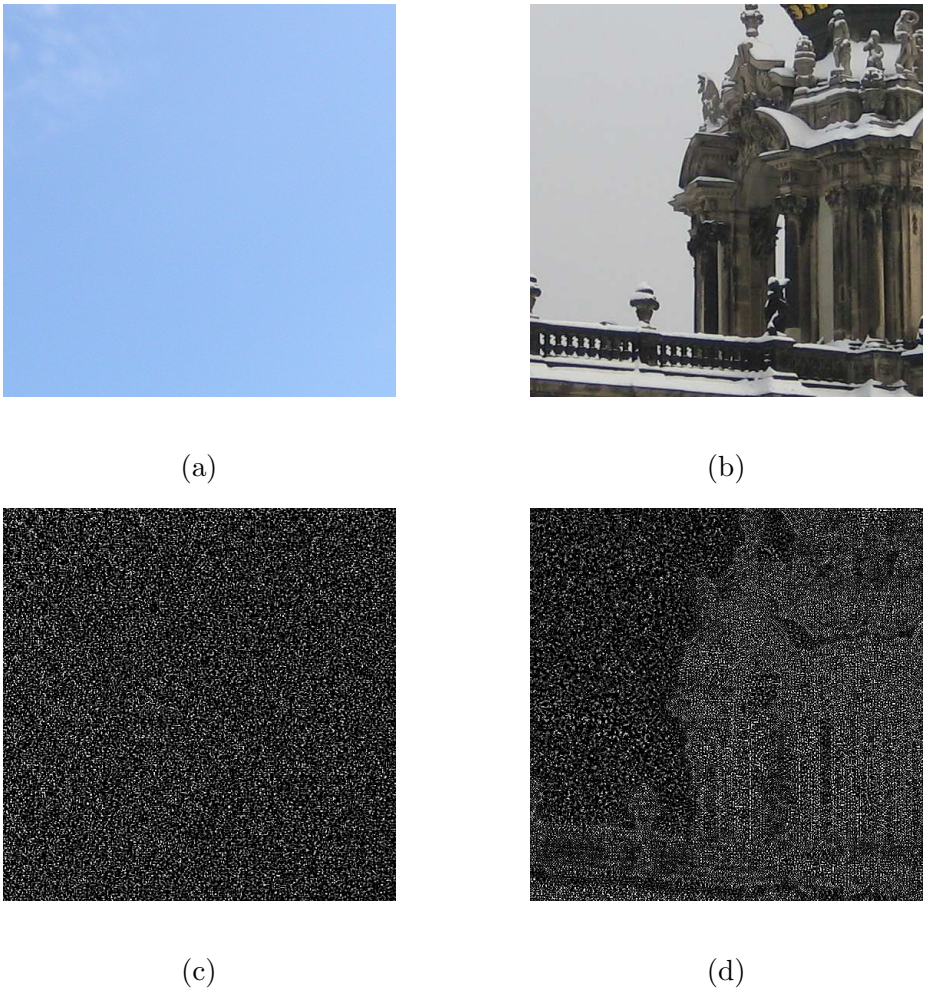


Figure 1.2: (a) A blue sky image. (b) An image with strong scene details. (c) The SPN extracted from the image (a). (d) The SPN extracted from the image (b). Note the intensity of (c) and (d) has been down scaled 3 times for visualization purpose.

are the SPNs extracted from Fig. 1.2 (a) and (b) by using the method proposed in [6], receptively. Compared to Fig. 1.2 (b), Fig. 1.2 (a) has much fewer scene textures so that its corresponding SPN (Fig. 1.2 (c)) is much less contaminated by scene textures. Therefore, images with smooth regions help to obtain better SPNs. However, in real-world environments, we may face the problem that the camera to be identified is absent. For example, a person is suspected of taking images with some illegal content (e.g., child pornographic, terror related) while he/she has

already abandoned or hidden his/her camera. Under this circumstance, there is no smooth images available for the police to estimate a clean reference SPN for the missing camera. But it is highly conceivable that the images from the suspect's Facebook or Flickr accounts are probably taken by the missing camera so that the police can estimate an SPN from such images to represent the missing camera. If any SPNs from the images with illegal content are found to be matched with this estimated SPN, the police can convict this suspect guilty. However, the images from Facebook or Flickr are very likely to be natural images with varying scene details. As a result, the SPN extracted from such images might be severely contaminated, which might lead to a false matching result. Therefore, a challenging problem is that: is it possible to estimate a trustworthy SPN from images with varying scene details so as to represent the missing camera?

1.4.2 Source Camera Identification in Large Database

Another challenging problem relates to the high computational complexity of a SPN-based SCI systems. This problem occurs due to the high dimensionality and random nature of the SPN fingerprints. Here we provide an example to explain this problem. A person is apprehended for suspiciously taking pictures of children near an elementary school, while he/she claims he/she is an amateur photographer pursuing a hobby. The police plans to estimate a reference SPN for his/her camera and search a large database of known child pornography images to check whether any existing clusters in this database match with this reference SPN. Ideally, this problem can be solved by matching this estimated SPN against all the SPNs of the clusters in the database, but the process would require a linear search over the

whole database. Considering the fact that most images today have more than 10^6 pixels and there can be tens of thousands of SPN fingerprints in the database, thus making the computational complexity of a direct search prohibitively high. In this case, the challenging part is that is it possible to solve this problem more efficiently? Although the SPN-based technique has been well studied by the research community, relatively fewer works are devoted to address this problem. Thus, research into the task of performing source camera identification more efficiently is very important and necessary.

1.5 Objectives of Thesis

In the view of the above-mentioned challenges, in this thesis, we explore the following two scenarios. The first scenario is effective source camera verification with reference images corrupted by scene details. As mentioned above, the reference SPN estimation is one of main stages in the framework of SPN-based source device identification. Although there have been several studies dedicated to improving the performance of SPN based source camera identification, an efficient method for estimating the reference SPN from images with scene details is still lacking. In order to address this problem, we propose a novel approach for estimating reliable reference SPN from natural images so as to improve the performance of source camera verification. In addition, we consider the situation that the number of reference images taken by the same camera is inadequate ($N < 50$), which is a case most of the current works do not take into account. Experimental results show that this method achieves better performance than the schemes based on the averaged reference SPN, especially when only a few reference images are used.

The second scenario is efficient source device identification in large database. Apart from identification accuracy, efficiency is also an important aspect of a source device identification system, especially when a sizeable database is involved. Intuitively, there are two ways to reduce the computational complexity. The first one is to organize and index the large database into a search tree so that there would a smaller number of SPN matching to be done. Another one is to find a compact representation of normal sized SPN so that even a linear search over the whole database can be performed. In the literature, some efforts have been made in these two directions. However, while many methods have improved the efficiency, they undesirably decrease the identification accuracy at the meantime. In the light of this limitation, we aim at improving the computational efficiency of source camera identification without degrading the identification accuracy. We employ the concept of PCA denoising [26–28] in the task of SPN-based source camera identification. An effective feature extraction algorithm based on this concept is proposed to extract a small set of components from the noisy SPN, which tends to carry most information of the true SPN signal. In order to extract components that better represent the true SPN signal rather than other noises, a training set construction procedure is proposed to facilitate a more accurate estimation of the feature extractor. In order to further improve the performance of this framework, two enhancement methods are introduced. Given the fact that investigators normally have the class label of the reference images in database, the first enhancement method is proposed to take the advantage of the label information of the reference images to better separate different classes and further reduce the dimensionality. Secondly, in real-world applications images taken by new cameras may be added to the database. In this

case, it is infeasible to re-conduct training every time when a new image arrives. To overcome this limitation, we propose an extension based on Candid Covariance-free Incremental PCA (CCIPCA) to incrementally update the feature extractor according to the received images.

However, the performance of the PCA-based feature extraction method degrades when there are some unwanted interferences contained by the training set, such as the artifacts introduced by scene details, CFA interpolation and JPECG compression. It is because the eigenvectors that generated from the training process can be corrupted by these interferences. Some leading eigenvectors are very likely to represent the interfering components rather than the true SPN signal. Moreover, it is difficult to locate or remove the corrupted eigenvectors from the feature space. Accordingly, it is difficult to train a reliable feature extractor by using a noisy training set. To address this problem, we propose a camera identification framework based on the random subspace method and majority voting. The experimental results show that this proposed solution is able to suppress the impact of unwanted interferences, consequently enhancing the identification accuracy. The main **contributions** are summarised in details as follows:

1. We propose a measurement based on the smoothness and brightness to evaluate the quality of each SPN block for the reference SPN estimation. Based on this measurement, a reference SPN estimator is proposed to address the problem that the reference images are contaminated by the scene details [29]. Instead of treating each SPN block equally for the reference SPN estimation, we weight each SPN block according to its quality. We also consider the situation that the number of reference images from each camera is inadequate ($N < 50$), which is

a case most of the current works do not take into account.

2. We employ the concept of PCA denoising in the task of source camera identification. A PCA-based feature extraction (PCAFE) algorithm is proposed to address the issue of prohibitively computational complexity caused by the high dimensionality of SPN [30, 31]. In order to extract components that better represent the true SPN signal rather than other noises, a training set construction method is also proposed to minimize the impact of the interfering artifacts in the training set. In addition, we proposed an extension based on the discriminate analysis method to take the advantage of the label information of the reference images, so as to better separate different classes and further reduce the dimensionality.

3. We propose a method based on CCIPCA [32] for incrementally updating the eigenvectors of the PCA-based feature extractor according to the new received images. It is an extension of the aforementioned PCAFE method, which is used to address the costly computation of re-conducting training caused by PCAFE whenever a new image arrives.

4. We point out the performance of PCAFE decreases when the training set is noisy and less representative. To address this problem, we propose an ensemble solution based on the random subspace method (RSM) [33] for the SPN-based source camera identification. This method can improve the identification accuracy by combining a large number of weak identifiers in the decision level (i.e., by using the majority voting).

1.6 Thesis Outline

Chapter 1 provides a brief review of varying techniques used for source camera identification, which can be roughly divided into three categories: metadata, digital watermarking and the intrinsic characteristics of digital cameras. Among these techniques, the advantages of using SPN to solve the source camera identification are briefly described. This background knowledge is important for further discussions in this thesis. The next chapter depicts the three main stages of the SCV and SCI system, and summarizes the related works which have been devoted in them. Chapter 3 through to 6, the main contributions to the field are elaborated. These include reducing the impact of scene details in reference SPN estimation (Chapter 3), representing SPN in a compact manner (Chapter 4), feature extractor with incremental update capability for fast source camera identification (Chapter 5) and the random subspace method in source camera identification (Chapter 6). Chapter 7 summarises the achievements of this thesis and presents some future research directions.

1.7 List of Publications

We provide the publication list for my PhD research on SPN-based source camera identification as follows:

1. **R. Li**, and C.-T. Li, Y. Guan, "Incremental Updating Feature Extraction for Camera Identification", in Proc. IEEE International Conference on Image Processing (ICIP), Quebec city, Canada, September 27-30, 2015.

Table 1.1: Thesis chapters and the corresponding publications.

Thesis Chapters	**Papers**	**Content**
Chapter 3	Paper 5	Context adaptive reference SPN estimator
Chapter 4	Paper 3, 4	PCA-based feature extraction
Chapter 5	Paper 1	Incrementally updated feature extraction
Chapter 6	Paper 2	RSM-based SCI system

2. **R. Li**, C. Kotropoulos, C.-T. Li, and Y. Guan, "Random Subspace Method for Source Camera Identification", in Proc. IEEE International Workshop on Machine Learning for Signal Processing, Boston, USA, Sept. 17-20, 2015.

3. **R. Li**, and C.-T. Li, Y. Guan,"A Compact Representation of Sensor Fingerprint for Camera Identification and Fingerprint Matching", in Proc. IEEE International Con- ference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, April 19-24, 2015.

4. **R. Li**, and Y. Guan, C.-T. Li, "PCA-based Denoising of Sensor Pattern Noise for Source Camera Identification", in Proc. IEEE China Summit&International Conference on Signal and Information Processing, Xi'an, China, July 9-13, 2014.

5. **R. Li**, and Y. Guan, C.-T. Li, "A Reference Estimator based on Composite Sensor Pattern Noise for Source Device Identification", in Proc. IS&T/SPIE Conference on Media Watermarking, Security, and Forensics, San Francisco, USA, February 2 - 6, 2014.

Some chapters of this thesis (i.e., Chapters 3-6) are related to the aforementioned papers, as listed in Table 1.1.

Chapter 2

Literature Review

In this chapter, we firstly review image acquisition process of an ordinary digital camera so as to better explain what SPN is and why SPN can be applied as a camera *fingerprint* to solve the source camera verification and identification problem. Generally, both processes of source camera verification and identification consist of three main stages: SPN extraction, reference SPN estimation and SPN matching. In Section 2.2, these three steps are described in details and relevant approaches proposed for these three steps are also reviewed. In Section 2.3, we introduce some common performance metrics for evaluating the performance of the SPN-based SCV or SCI system. Finally, we discuss the limitations of the current approaches and analyse the two problems mentioned in Chapter 1.

2.1 Image Acquisition Process

Image acquisition is actually a process of converting the incident light into a digital signal representation of the scenery. Typically, the process of acquiring an image

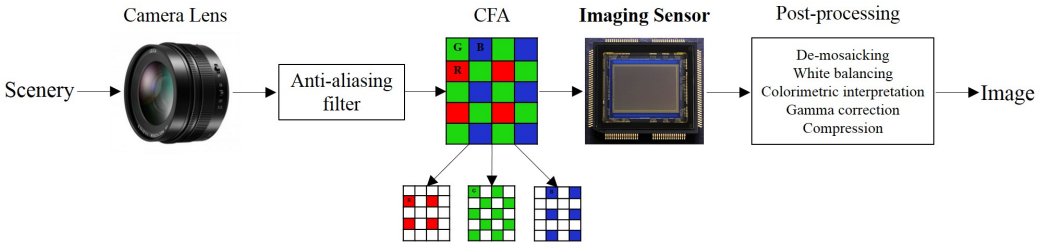


Figure 2.1: A simplified depiction of an imaging pipeline within a digital camera.

with an ordinary digital camera is illustrated by the diagram of Fig. 2.1 [34]. The light from the scene entering a camera is first filtered by the lens and an anti-aliasing filter [35] before reaching the imaging sensor. The most important and expensive part of a digital camera is the imaging sensor. Since the elements of imaging sensor are monochromatic, each sensor element can only capture one colour value [36], such as red (R), green (G), or blue (B). As a result, the imaging sensor can not separate colour information. Therefore, a colour filter array is required to be built above the imaging sensor so as to help with separating the colour information. After colour filtering, the light is captured by the elements of imaging sensor and converted into individual pixels that comprise the image. Later, a demosaicing operation, which is tailored for the type of colour filter array by the camera manufacturer, takes place to calculate the missing colour values for each pixel so as to generate a full-colour image (with intensities of all three primary colours at each pixel). This is followed by a sequence of post-processing operations, such as white balancing operation, colourimetric interpretation, and gamma correction [37]. At the end, the generated image is compressed and saved in the camera's memory.

Among various stages of image acquisition process described above, the sensor imaging operation is the most important one, and it is the stage where sensor

pattern noise is produced. In this stage, each element of the imaging sensor would capture the incident light and convert it into an individual digital pixel. Due to the non-homogeneity of silicon wafer, normally different sensor elements have different sensitivity to light. As a result, even if the imaging sensor takes an image of an absolutely evenly lit scene, the resulting image would still exhibit slight variations in the intensity between the individual pixels [38]. Such variations between individual pixels form a noise pattern, which is called the sensor pattern noise. As reported in [6], if a camera takes multiple images of exactly the same scenery, the SPN patterns left in these images would stay approximately similar. It indicates that every image or video taken by the same sensor would exhibit the same SPN pattern. In addition, because of the imperfections during the sensor manufacturing process, even two imaging sensors made from the same silicon wafer would exhibit uncorrelated SPN patterns. As a result, the SPN produced by each sensor is unique. Due to these proprieties, the SPN pattern can be viewed as a unique *fingerprint* of a digital camera. Its uniqueness allows SPN to distinguish not only different camera models of the same brand, but also individual cameras of the same model.

Sensor pattern noise consists of two main components: fixed pattern noise (FPN) and photo-response non-uniformity (PRNU) noise [39]. FPN refers to pixel-to-pixel differences when the sensor array is not exposed to light. It is primarily caused by dark currents. FPN is an additive noise, thus some middle-end and high-end cameras can automatically suppress this noise by subtracting a dark frame from every image they take [24]. On the other hand, PRNU noise is the discriminative part of SPN. The PRNU noise is a multiplicative noise [6], and its amplitude depends on the brightness of the observed image. Generally speaking, the brighter

the observed image is, the stronger the PRNU noise would be. However, the PRNU noise cannot be present in completely saturated images or image areas because the pixels were filled to their full capacity, thus a saturated image would carry no information about the PRNU noise. For the same reason, the PRNU noise is very weak in dark areas, leaving the FPN as the dominant component of the SPN.

2.2 SPN Extraction

As mentioned in Chapter 1.3, in order to verify whether a query image was taken by a specific camera using SPN, three main steps are required: SPN extraction, reference SPN estimation and SPN matching. In this section, we first introduce the SPN extraction step, the purpose of which is to extract the SPN from the query image. In [6], Lukas *et al.* modelled the output of a camera in a simplified form:

$$\mathbf{I} = (1 + \mathbf{K})\mathbf{I}^{(0)} + \mathbf{\Theta} = \mathbf{I}^{(0)} + \mathbf{I}^{(0)}\mathbf{K} + \mathbf{\Theta}, \tag{2.1}$$

where $\mathbf{I}^{(0)}$ is the sensor output in the absence of noise. The multiplicative factor $\mathbf{K}$ refers to the PRNU factor ($\mathbf{K} < 1$). $\mathbf{I}^{(0)}\mathbf{K}$ thereby represents the discriminative part of SPN, which is the signal of interest. $\mathbf{\Theta}$ is the composite of independent random noise components, which includes the dark current, shot noise, read-out noise, and quantization noise. In the rest of this thesis, matrices are shown in capital bold; vectors are presented in bold lower-case; operations and variables are in element-wise.

In order to extract the signal of interest $\mathbf{I}^{(0)}\mathbf{K}$ from the observed data $\mathbf{I}$, the host signal $\mathbf{I}^{(0)}$ should be removed. However, the noiseless image $\mathbf{I}^{(0)}$ is generally unknown as most camera manufacturers do not allow user to access the raw sensor

output. Nevertheless, it is possible to obtain an approximation to the noiseless image $\mathbf{I}^{(0)}$ by using a denoising algorithm, $\hat{\mathbf{I}}^{(0)} = F(\mathbf{I})$, where $\hat{\mathbf{I}}^{(0)}$ is a denoised version of the image $\mathbf{I}$, F indicates the denoising algorithm. Therefore, SPN can be estimated as the noise residual between the observation $\mathbf{I}$ and its denoised version $\hat{\mathbf{I}}^{(0)}$. For example, by subtracting the denoised version $\hat{\mathbf{I}}^{(0)}$ from the observation $\mathbf{I}$, we can extract an approximation of SPN as

$$\begin{aligned}\mathbf{X} &= \mathbf{I} - F(\mathbf{I}) = \mathbf{I} - \hat{\mathbf{I}}^{(0)} \\ &= \mathbf{I}^{(0)} - \hat{\mathbf{I}}^{(0)} + \mathbf{I}^{(0)}\mathbf{K} + \boldsymbol{\Theta} = \mathbf{I}\mathbf{K} + \boldsymbol{\Xi},\end{aligned} \tag{2.2}$$

where $\mathbf{X}$ is the noise residual where the true SPN signal is present. The noise $\boldsymbol{\Xi}$ is a sum of $\boldsymbol{\Theta}$ and the additional noise terms introduced by the denoising filter. From this equation, we can deduce that the closer the denoised version $\hat{\mathbf{I}}^{(0)}$ is to the noiseless image $\mathbf{I}^{(0)}$, the purer SPN signal would be extracted in the noise residual $\mathbf{X}$. Therefore, the performance of an SPN extractor is primarily determined by the choice of the denoising algorithm F.

2.2.1 Denoising Algorithm

Since the denoising algorithm plays an important role at determining the quality of the extracted SPN, the potential choices for the denoising algorithm is worth further discussing. According to the works proposed by the previous researchers, there are three most popular techniques: Mihcak filter [40], SPN predictor based on context adaptive interpolation (PCAI) [11, 12] and block-matching and 3D filtering (BM3D) [8].

Mihcak Denoising Filter

In [6], Lukas *et al.* pointed out that SPN is a high-frequency signal in an image, thus they proposed to transform the noisy image **I** into wavelet transform domain and apply a wavelet-based denoising filter (Mihcak denoising filter [40]) to extract the SPN components from the high frequency wavelet coefficients of **I**. Since this method [6] was the firstly proposed in literature for SPN extraction, we refer it as "Basic" method in this thesis.

As mentioned in [6], the authors had tested with several denoising filters and eventually decided to use the Mihcak denoising filter as it provided the best results. The underlying hypothesis of this method is that the high-frequency wavelet coefficients of a noisy image can be modelled as an additive mixture of a locally stationary i.i.d. signal with zero mean (noise-free image) and a stationary white Gaussian noise (WGN) signal with variance σ_0^2 (i.e. the SPN). Based on this hypothesis, the Mihcak denoising filter is built in three steps. In the first step, the input noisy image is processed by the wavelet decomposition. In the second step, the local image variance is estimated. Finally, the local Wiener filter is used to obtain an estimate of the denoised image in the wavelet domain. The individual steps are described as follows:

Step 1. *Wavelet decomposition.* Calculate the fourth-level wavelet decomposition of the noisy image with the 8-tap Daubechies QMF. We describe the procedure for one fixed level (it is executed for the high-frequency bands for all four levels). Denote the vertical, horizontal, and diagonal subbands as $h(i,j)$, $v(i,j)$, $d(i,j)$, where (i,j) runs through an index set J that depends on the

decomposition level.

Step 2. *Local variance estimation.* In each subband, estimate the local variance of the noise-free image for each wavelet coefficient using the Maximum A-Posteriori Probability (MAP) estimation [41] for four sizes of a square $m \times m$ neighbourhood N_m (where $m \in \{3, 5, 7, 9\}$), such as

$$\hat{\sigma}_m^2(i,j) = max\left(0, \frac{1}{m^2} \sum_{(i,j) \in N_m} h^2(i,j) - \sigma_0^2\right), (i,j) \in J. \tag{2.3}$$

The local variance of the noise-free image $\hat{\sigma}^2(i,j)$ is obtained by taking the minimum of the four variances

$$\hat{\sigma}^2(i,j) = min(\sigma_3^2(i,j), \sigma_5^2(i,j), \sigma_7^2(i,j), \sigma_9^2(i,j)), (i,j) \in J. \tag{2.4}$$

Step 3. *Wiener filtering.* In each subband, denoise the wavelet coefficients by using a pixel-wise adaptive Wiener filter based on the estimated local variance from the neighbourhood of each pixel, such as

$$h_{den}(i,j) = h(i,j) \frac{\hat{\sigma}^2(i,j)}{\hat{\sigma}^2(i,j) + \sigma_0^2}, \tag{2.5}$$

where $\hat{\sigma}^2(i,j)$ is the estimated variance of the noise-free image and σ_0^2 is the variance of the WGN signal. Similarly, $v_{den}(i,j)$, and $d_{den}(i,j)$ can be obtained in the same way, where $(i,j) \in J$. By repeating Steps 1-3 for each level and each colour channel, a denoised image can be finally obtained in the spatial domain by applying the inverse wavelet transform to the denoised wavelet coefficients.

Notice that the parameter σ_0 is still unknown. As reported in [6], σ_0 is normally set to $\sigma_0 \in \{1, ..., 6\}$ so as to better extract the SPN signal, and within this range, the choice of σ_0 has a relatively low impact on the final identification results. But for the images with large noise components, such as images with strong scene details and images which are highly compressed, setting σ_0 to a relatively large value would make sure that the filter extracts a substantial part of the SPN. For this reason, we have chosen $\sigma_0 = 4$ in all the following experiments of this thesis. Based on the Eq. (2.2), this Basic method can be simply formulated as follows

$$\mathbf{X} = DWT^{-1}\{DWT(\mathbf{I}) - F[DWT(\mathbf{I})]\}, \tag{2.6}$$

where DWT is the discrete wavelet transform, and DWT^{-1} is the inverse wavelet transform. F is the Mihcak denosing filter, which is constructed in the wavelet domain using the Wavelab package [42].

PCAI Predictor

In [12], Wu *et al.* proposed an edge adaptive predictor based on the context adaptive interpolation (PCAI) to extract SPN. Since the scene texture is the most serious source that contaminates the true SPN signal, this PCAI method was designed mainly to suppress the impact of scene texture. The context adaptive interpolation (CAI) method [43] is an interpolation method which can predict the texture within an image by using the local neighbouring information. However, subtle signal, such as SPN, is not very likely to be accurately predicted in the output by this method. Thus the difference between the predicted value and actual value can better suppress

the impact of image edges while preserving the SPN components at the same time.

- **The CAI interpolation method.** Assume that p is the central pixel and $\mathbf{t} = \{n, s, e, w\}^T$ is the vector of neighbouring pixels. The CAl interpolation method would predict an approximation $\hat{p}$ for the pixel value p according to its neighbouring information $\mathbf{t}$. More specifically, the CAl method firstly scans the whole image and classifies each pixel (according to its local region) into four categories: smooth, horizontally-edged, vertically-edged and other. In the smooth region, a mean filter is used to estimate the central pixel; in edged regions, the interpolation is done along the edge; otherwise a median filter is applied. The predicted pixel value $\hat{p}$ can be formulated as follows [43]

$$\hat{p} = \begin{cases} mean(\mathbf{t}), & (max(\mathbf{t}) - min(\mathbf{t}) \leq 20) \\ (n+s)/2, & (|e-w| - |n-s| > 20) \\ (e+w)/2, & (|n-s| - |e-w|) > 20) \\ median(\mathbf{t}), & (otherwise). \end{cases} \tag{2.7}$$

In Eq. (2.7), the central pixel, which is predicted according to different types of edge regions, is classified by the four-neighbouring pixel values with a threshold value. According to [12], this threshold is set to be 20 via the extensive experiments. In [11], the CAI method is extended by using of the eight-neighbouring pixels (including four diagonally-edged pixels), which is called as "CAI8". By doing so, the authors claimed that the predicted result $\hat{p}$ would have less prediction error [11].

- **The PCAI predictor.** In [11], an edge adaptive SPN predictor is proposed based on this CAI8 method, which is referred to as "PCAI8" for short in this thesis.

This PCAI8 predictor is built in two steps as follows.

Step 1. Firstly, the CAI8 method is applied as the desnoising filter F in Eq. (2.2). Then, the difference $\mathbf{D}$ between the predicted result and original image can be calculated in the spatial domain as follows

$$\mathbf{D} = \mathbf{I} - CAI(\mathbf{I}), \tag{2.8}$$

where $CAI(\mathbf{I})$ is a pixel-wise prediction of the original image $\mathbf{I}$. The CAI method can only predict the scene texture of $\mathbf{I}$ but not the SPN components. Therefore, the scene texture would be suppressed in the difference $\mathbf{D}$ while the SPN components would be well preserved.

Step 2. In order to further suppress the impact of the scene texture and extract a more accurate SPN, a pixel-wise adaptive Wiener filter is then preformed as follows [11]

$$\mathbf{X}(i,j) = \mathbf{D}(i,j)\frac{\sigma_0^2}{\sigma^2(i,j) + \sigma_0^2}, \tag{2.9}$$

where $\mathbf{X}$ is the eventual output of the PCAI8 predictor and the noise residual that contains the SPN components. σ^2 indicates the estimated local variance of the noise residual $\mathbf{D}$ and σ_0^2 is the variance of the WGN signal, i.e. the SPN. The local variance is estimated by using the MAP estimation, which is similar to Eq. (2.4) and (2.5). The parameter σ_0 is also set to be 4 in order to extract a consistent level of the SPN.

Since the CAI method can predict texture accurately according to different local regions, the PCAI method is usually superior to Mihcak filter on extracting SPNs

from the images with strong scene details [43]. However, due to the pixel-wise interpolation, this method is more computationally complex.

BM3D Denoising Filter

In [8], Chierchia *et al.* proposed to replace the Mihcak filter with a more recent technique, namely the block-matching and 3D filtering (BM3D) [44], in order to extract SPN. BM3D works through grouping 2D image patches with similar structures into 3D arrays and collectively filtering the grouped image blocks. The sparseness of the representation due to the similarity between the grouped blocks makes it capable of better separating the noiseless image and noises. This filter is constructed using two steps as follows:

Step 1. *Basic estimate.* The input noisy image is processed by successively extracting the reference blocks from the image (in a sliding-window manner, *e.g.,* 8×8).

a) *Block-wise estimates.* For each reference block in the noisy image, find the blocks that are similar to the currently processed one and then stack them together in a 3D array (group). Then, apply a 3D transform to the formed group, attenuate the noise by a hard-thresholding of the transform coefficients, invert the 3D transform to produce estimates of all grouped blocks, and return the estimates of the blocks to their original positions.

b) *Aggregation.* Compute the basic estimate of the true-image by a weighted averaging of all the obtained block-wise estimates that are overlapping.

Step 2. *Final estimate.* Using the basic estimate, perform improved grouping and

collaborative Wiener ltering.

a) *Block-wise estimates.* For each block, use block-matching within the basic estimate to find the locations of the blocks similar to the currently processed one. Using these locations to form two groups (3D arrays), one from the noisy image and one from the basic estimate. Then, apply a 3D transform on both groups. Perform Wiener filtering with $\sigma_0 = 4$ on the noisy one using the variance of the basic estimate as the true variance. Produce estimates of all grouped blocks by applying the inverse 3D transform on the filtered coefficients and return the estimates of the blocks to their original positions.

b) *Aggregation.* Compute a final estimate of the true image by aggregating all of the obtained local estimates using a weighted average.

Generally speaking, among the three aforementioned SPN extraction methods, the BM3D method can slightly outperform the Basic and PCAI8 method on SPN extraction. For the example in Fig. 2.2, (a) shows an image with strong scene details. (b), (c) and (d) are the noise residuals extracted from the image of (a) by using the Basic, PCAI8 and BM3D method, respectively. By comparing (b), (c) and (d), we can see that the BM3D method can better suppress the impact of scene details than the Basic and PCAI8 counterparts. It is because both of the Mihcak filter and PCAI8 predictor estimate the variance of noise-free image by using only the local neighbourhood information, while the estimate of BM3D filter is obtained by collaboratively aggregating the estimates from multiple non-local blocks. The denoising output $F(\mathbf{I})$ from BM3D filter is, therefore, closer to the true noise-free image $\mathbf{I}^0$. The source code for this BM3D filter is available online at [45].

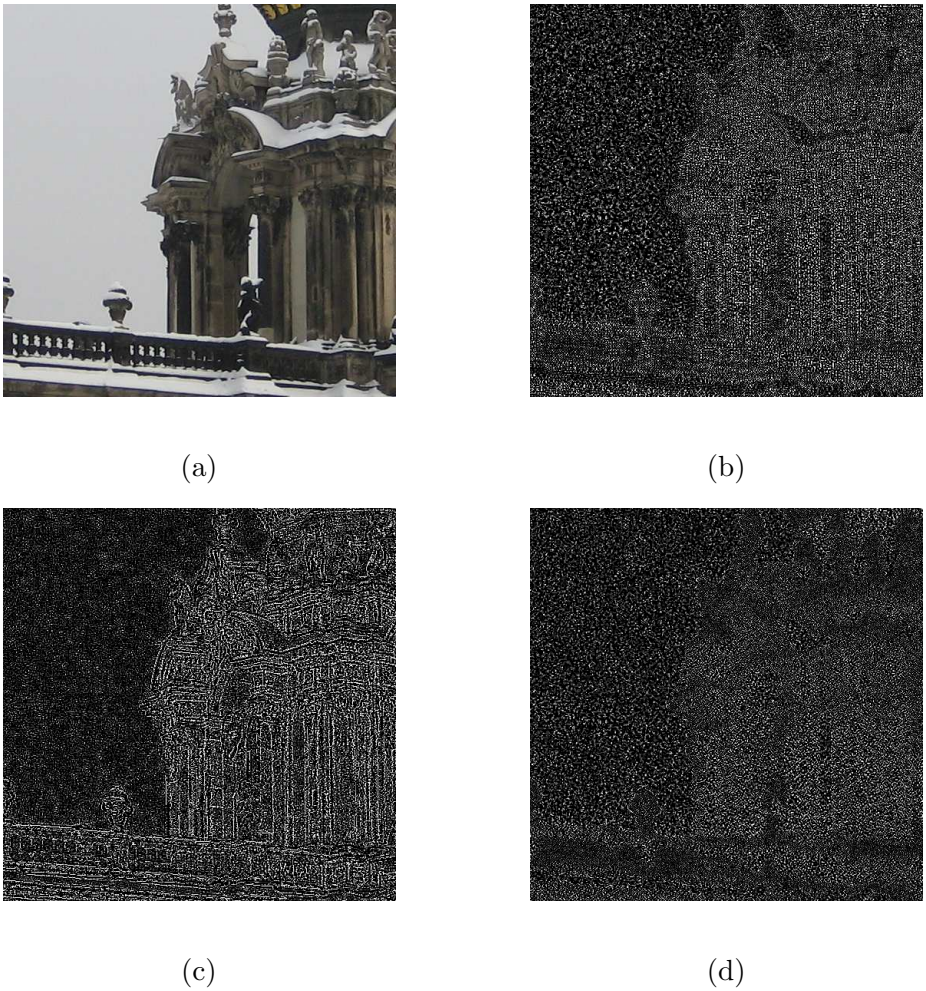


Figure 2.2: (a) An image with strong scene details. (b) The noise residual extracted from the image (a) using the Basic method. (c) The noise residual extracted from the image (a) using the PCAI8 method. (d) The noise residual extracted from the image (a) using the BM3D method. Note the intensity of (b), (c) and (d) has been downscaled 3 times for visualization purpose.

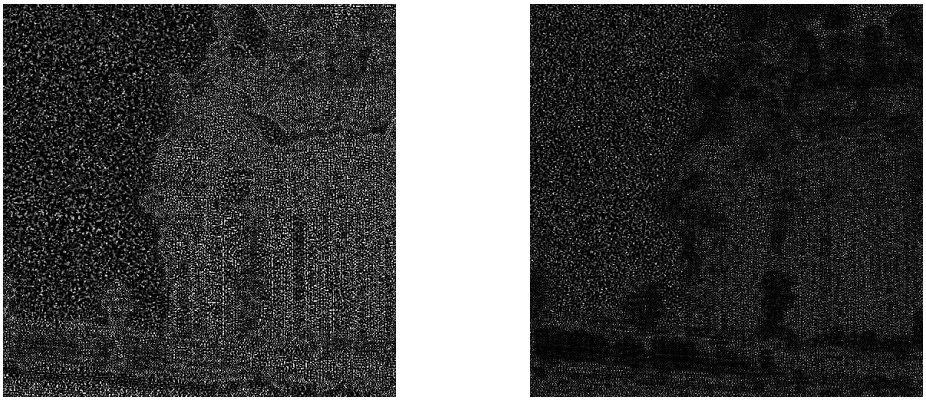
Moreover, there are some SPN extraction methods proposed in allusion to eliminating specific contaminations. For example, Li *et al.* [10] introduced a colour-decoupled PRNU (CD-PRNU) extraction method to prevent the interpolation noise from propagating into the physical components. They extracted the PRNU noise patterns from each colour channel and then assembled them to get the more reliable

CD-PRNU. In [46], Al-Ani *et al.* developed another image denoising algorithm for SPN estimation. The authors claimed that involving a large number of pixels at the denoising operation to approximate a single pixel would result in a considerable level of unwanted correlation between neighbouring pixels in the extracted noise residual. As a result, the obtained noise residual cannot well manifest the characteristic of the true SPN components. In order to suppress the correlation between neighbouring pixels in the extracted noise residual, they proposed to produce a noise estimate of SPN at a pixel via subtracting this pixel by one adjacent pixel which has a close value.

2.2.2 SPN Enhancement

As shown in Eq. (2.2), the extracted noise residual contains not only the true SPN components but also some unwanted interferences, such as random noise components, scene details, CFA artifacts, etc. Thus it leaves room for further enhancement.

In [9], Li pointed out that the contaminations from scene details is the most serious one among these interferences, the magnitude of which is far greater than that of true SPN signal. Since the scene details also account for the high-frequency components of $\mathbf{I}$, it would mix with the true SPN signal and be extracted into the noise residual at the same time. As shown in Fig. 2.2, the scene textures appearing in Fig. 2.2 (a) propagate through the three SPN extraction methods into the noise residual and contaminate the true SPN signal. Although BM3D filter is reported that can better suppress the impact of scene details, there are still some scene textures left in the noise residual. Involving such interferences in the noise



(a) (b)

Figure 2.3: (a) The noise residual extracted by using the Basic method. (b) The enhanced version of (a) by using the Model 5 in [9]. Note the intensity of (a) and (b) has been down scaled 3 times for visualization purpose.

residuals would lead to two noise residuals being correlated even though they are derived from different cameras. As a result, it would increase the false identification rate. To overcome this limitation, Li [9] proposed an SPN enhancer to attenuate the impact of scene details so as to enhance the true SPN signal in noise residual. The hypothesis underlying his SPN enhancer is that the stronger a signal component in $\mathbf{X}$ is, the less trustworthy the component should be, and thus it should be attenuated. According to this hypothesis, the author provided 5 spatial-domain based enhancing models aiming at assigning less significant weighting factors to strong components of $\mathbf{X}$ so as to attenuate the interference of scene details. Here is an example shown in Fig. 2.3, where Fig. 2.3 (a) is the noise residual extracted by using the Basic method, and Fig. 2.3 (b) is the enhanced version of (a) by using the Model 5 in [9]. We can clearlys see that the trace of scene details left in Fig. 2.3 (a) has been significantly suppressed in Fig. 2.3 (b). It suggests that this enhancement scheme can be applied as the post-processing method after the SPN extraction so as to

enhance the purity of the SPN components in the noisy noise residual.

In [47], Lawgaly *et al.* proposed an enhancement method based on the Unsharp Masking technique [48], which aims to amplify the high frequency content of the SPN in images. This method can strengthen the SPN present in images and consequently enhance the SPN components in the estimated noise residuals. It can be applied as a pre-processing method performed before the SPN extraction.

Moreover, there are some methods proposed to suppress the contamination caused by JPEG compression. In JEPG compression, non-overlapping 8×8 pixels blocks are coded with the discrete cosine transform (DCT) independently [49]. As a result, aggressive JPEG compression would cause blocky artifacts in the extracted noise residual. In [50, 51], Alles *et al.* proposed a method to suppress the JPEG block artifacts by averaging neighbouring 8×8 pixels block into one macro element on both the query noise residual and reference SPN. By doing so, the sizes of the query noise residual and reference SPN are decreased. While this method can attenuate the impact of JPEG blocky artifacts, it also suppresses the true SPN components. In [52], Chen *et al.* proposed a more robust method to remove the JPEG blocky artifacts by transforming the extracted noise residual into the discrete Fourier transform (DFT) domain and suppressing the Fourier coefficients with extremely larger magnitude. By doing so, they claimed that not only JPEG blocky artifacts but also other artifacts which manifest themselves as peaks in the Fourier domain (e.g., artifacts due to colour filter array interpolation and other hardware or software operations) can be suppressed.

Certainly, the methods for SPN extraction and SPN enhancement can be combined for performance gains. For instance, one can apply BM3D or PCAI8

algorithm to extract the raw noise residual, and then enhance it with the help of Li's models [9].

2.3 Reference SPN Estimation

The reference SPN estimation step aims at estimating a reference SPN pattern for the camera in question. In [6], Lukas first proposed to estimate the reference SPN $\mathbf{R}$ for a camera by averaging N noise residuals extracted from the reference images taken by that camera, such as

$$\mathbf{R} = \sum_{i=1}^{N} \mathbf{X}_i / N. \tag{2.10}$$

Generally speaking, random noises presented in different images are normally quite different, while the true SPN components will be the same as long as these images are taken by the same camera. Therefore, the random noise components (e.g. shot noise, read-out noise and quantization noise) can be averaged out in $\mathbf{R}$ while the true SPN components are accumulated. It is obvious that the larger the number of images N is, the more random noise components we can suppress. As suggested in [6], it is optimal to set $N \geqslant 50$. It is worth noting that if the camera to be identified is available, it is better to use the low-variation images such as blue sky and flat field images for reference SPN estimation, so that we can actively avoid the contaminations caused by scene details. By doing so, we actually incorporate our prior knowledge of SPN to refine the estimated result.

In [7], Chen *et al.* proposed a maximum likelihood method to estimate the reference SPN. As shown in Eq. (2.2), the discriminative part of SPN, i.e., PRNU,

is a multiplicative noise, thus the goal of this method is to estimate the reference PRNU factor $\mathbf{K}$ for a camera. They assume that the pixels of the noise term $\mathbf{\Xi}$ is zero-mean Gaussian noise with variance σ^2 and independent from the signal $\mathbf{IK}$. Let $\{\mathbf{I}_i\}_{i=1}^N$ be the reference images from camera C. For each image $\mathbf{I}_i$, $i = 1, 2, ..., N$, the Eq. (2.2) can be re-written as:

$$\frac{\mathbf{X}_i}{\mathbf{I}_i} = \mathbf{K}_i + \frac{\mathbf{\Xi_i}}{\mathbf{I}_i}, \quad \text{where} \quad \mathbf{X}_i = \mathbf{I}_i - F(\mathbf{I}_i). \tag{2.11}$$

Finally, the reference PRNU factor for camera C can be estimated by the Maximum Likelihood estimation:

$$\hat{\mathbf{K}} = \frac{\sum_{i=1}^N \mathbf{X}_i \mathbf{I}_i}{\sum_{i=1}^N \mathbf{I}_i^2}. \tag{2.12}$$

Now given a query image $\mathbf{I_q}$, this method [7] would regard the term $\mathbf{I_q}\hat{\mathbf{K}}$ as the reference SPN left in image $\mathbf{I_q}$ by camera C.

In [53], Hu *et al.* pointed out that the components of the reference SPN with larger values are more robust. Therefore, instead of using the full-size reference SPN, authors only select a small number of the largest components from it as the reference. Moreover, they also record the location information of these largest components so that they can select the corresponding components from the query SPN to perform a matching. This method works very well when the reference SPN is clean. However, its performance would degrade when the reference SPN is contaminated by scene details, because the large components in such a contaminated SPN is more likely to be associated with scene details rather than true SPN components.

To further refine the estimated reference SPN, Chen *et al.* [7] proposed two preprocessing operations, zero-mean (ZM) and the Wiener filter (WF) in the dis-

crete Fourier transform (DFT) domain, to remove the artifacts caused by camera processing operations from the reference SPN. The ZM procedure is to remove the artifacts introduced by CFA interpolation, row-wise and column-wise operations of sensors or processing circuits. The objective of the WF procedure is to suppress the visually identifiable patterns in the ZM processed signal. These two pre-processing operations can be summarized as follows

$$\mathbf{R}_{WF} = DFT^{-1}(DFT(\mathbf{R}_{ZM}) - F(DFT(R_{ZM}))), \tag{2.13}$$

in which DFT is the discrete Fourier transform, DFT^{-1} is the inverse discrete Fourier transform. F is the Wiener filter with the variance obtained as the sample variance of the magnitude of the $DFT(\mathbf{R}_{ZM})$.

In [54], another reference SPN pre-processing approach, namely the Spectrum Equalization Algorithm (SEA), is recently proposed to suppress the periodic artifacts, such as CFA interpolation artifacts, JPEG blocky artifacts and diagonal artifacts, from the estimated reference SPN. The authors claimed that the peaks appearing in the DFT spectrum are probably originated from the periodic artifacts and unlikely to be associated with the true SPN. Therefore, by detecting and suppressing the peaks in the DFT spectrum, the reference SPN of better quality can be obtained.

2.4 SPN Matching

Once both query SPN and reference SPN are obtained, the SPN matching step is finally performed to measure the similarity between the query SPN and the reference SPN so as to decide whether the query image was taken by the camera in question. This problem can be treated as a binary-hypothesis test with the two hypotheses [55], which are defined as

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

$$H_1 : \mathbf{X} = \mathbf{R} \text{(the query image is taken by the suspect camera)}$$

Then a correlation-based detector is used to make the decision between H_0 and H_1 by comparing the correlation $\rho(\mathbf{X}, \mathbf{R})$ to a pre-calculated threshold τ. The detector decides H_1 when $\rho \geqslant \tau$ and H_0 when $\rho < \tau$.

2.4.1 Similarity Measurement

For this type of problem, the normalized cross-correlation (NCC) is usually used as the similarity measurement (detection statistics) to measure the similarity between the query noise residual and the reference SPN, which is defined as:

$$\rho\left(\mathbf{X}, \mathbf{R}\right) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\mathbf{X}(i,j) - \overline{\mathbf{X}}\right)\left(\mathbf{R}(i,j) - \overline{\mathbf{R}}\right)}{\left\|\mathbf{X} - \overline{\mathbf{X}}\right\| \cdot \left\|\mathbf{R} - \overline{\mathbf{R}}\right\|}, \tag{2.14}$$

where $\overline{\mathbf{X}}$ and $\overline{\mathbf{R}}$ are the mean value of $\mathbf{X}$ and $\mathbf{R}$, $N \times N$ is the size of $\mathbf{X}$ and $\mathbf{R}$, and $\|\cdot\|$ is the L_2 norm. Later, Goljan *et al.* [56] pointed out that the normalized cross-correlation is sensitive to the influence of periodic noise contaminations, therefore they proposed another similarity measurement, peak-to-correlation energy (PCE)

ratio [57], to suppress periodic noise contamination, such as

$$PCE(\mathbf{X},\mathbf{R}) = \frac{C^2_{\mathbf{RX}}(0,0)}{\frac{1}{N^2-|A|}\sum_{(i,j)\notin A} C^2_{\mathbf{RX}}(i,j)}, \tag{2.15}$$

where C^2_{RX} is circular cross-correlation between the two fingerprints $\mathbf{X}$ and $\mathbf{R}$, A is a small area around $(0,0)$, and $|A|$ is the size of A. In the real-world applications, when the final decision will be served as a vital piece of evidence in a crime investigation, a low false positive rate (FPR) is usually required so as to ensure a low probability of wrong accusation. Therefore, Kang *et al.* [13] proposed a more sophisticated detection statistic, i.e., the correlation over circular correlation norm (CCN), to further reduce the FPR of a source camera identification system, such as

$$CCN(\mathbf{X},\mathbf{R}) = \frac{C_{\mathbf{RX}}(0,0)}{\sqrt{\frac{1}{N^2-|A|}\sum_{(i,j)\notin A} C^2_{\mathbf{RX}}(i,j)}}. \tag{2.16}$$

2.4.2 Detection Threshold

The accuracy of a SCV or SCI system is defined by true positive rate (TPR) and false positive rate (FPR). True positive rate is the probability of deciding H_1 while hypothesis H_1 is true, and false positive rate is the probability of deciding H_1 while H_0 is true. Investigators usually require an SCV or SCI system to have a sufficiently low FPR so as to ensure a low probability of wrong accusation in some real-world applications, such as child pornography cases. Therefore, the detection threshold τ is estimated according to a required FPR.

According to Eq. (2.2), we simplify and rewrite it into a vectorized form as:

$$\mathbf{x} = \hat{\mathbf{x}} + \xi, \tag{2.17}$$

where $\mathbf{x}$ indicates the noise residual, $\hat{\mathbf{x}}$ presents the true SPN signal and ξ is WGN. Following the assumptions of [58], we assume that the noise residuals to be matched are normally distributed random sequences with zero mean and unit variance such that $\mathbf{x} = \{x_i\} = \{\hat{x}_i + \xi_i\}$, $i = 1, ..., m$, where $\xi_i \sim \mathcal{N}(0, \sigma^2)$, $\hat{x}_i \sim \mathcal{N}(0, 1-\sigma^2)$ and m is the length of the vectorized noise residual. We also assume that SPNs of two different cameras are statistically independent. According to the central limit theorem (CLT) [59], the NCC values for independent vectors follow the Gaussian distribution. Therefore, from the independence of SPNs assumption, the distribution of the NCC values for non-matching SPNs (under hypothesis H_0) can be approximately estimated as a normal distribution with zero mean and variance equal to $1/m$, i.e., $\rho^{H_0} \sim \mathcal{N}(0, 1/m)$. Given a detection threshold, the false positive rate P_{fp} for SPN can be obtained via Neyman-Pearson hypothesis approach [60] as follows

$$P_{fp} = \mathcal{Q}(\tau\sqrt{m}), \tag{2.18}$$

where $\mathcal{Q}$ is the complementary cumulative density function of a normal random variable $N(0, 1)$. Given a required P_{fp}, the detection threshold can be obtained via an inverse operation as

$$\tau = \frac{\mathcal{Q}^{-1}(P_{fp})}{\sqrt{m}}, \tag{2.19}$$

where $\mathcal{Q}^{-1}$ is a scaled inverse error function.

2.5 Performance Metrics

In the scenario of SPN-based source camera verification, the receiver operating characteristic (ROC) curve is usually applied to evaluate the accuracy of a source camera identification system. The ROC curve is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied [61]. The curve is created by plotting the true positive rate against the false positive rate at various threshold settings.

- The *true positive rate* is the rate at which the comparison between two matched SPNs are correctly accepted by the system as the true match.
- The *false positive rate* is the rate at which the comparison between two non-matched SPNs are erroneously accepted by the system as the true match.

In [13], Kang *et al.* proposed an overall ROC curve which can evaluate an average performance of a source camera identification system on different cameras. To draw the overall ROC curve, the number of true positive decisions and false positive decisions are first recorded for each camera respectively. A true positive occurs if hypothesis H_1 is accepted when the query image is indeed taken by the camera in question. A false positive occurs if H_1 is accepted while the query image is not taken by the camera in question. These numbers are then summed up to calculate the true positive rate P_{tp} and false positive rate P_{fp}. If the numbers of images captured by each camera are exactly the same, we can simply calculate the overall TPR and overall FPR for a threshold as follows

$$P_{tp} = \frac{\sum_{i=1}^{c} D_{tp}^{i}}{T}, P_{fp} = \frac{\sum_{i=1}^{c} D_{fp}^{i}}{(c-1)T}, i = 1, 2, ..., c, \tag{2.20}$$

Table 2.1: Two typical contamination cases

Query	**Reference**	**Scenario**	**Application**
Contaminated	Uncontaminated	Proof of ownership, copyright protection	Child pornography, movie piracy
Contaminated	Contaminated	Law enforcement, police and security	Criminal and terrorist investigation, detection of networks for sharing images

where c is the number of cameras, T is the total number of query images, D^i_{tp} and D^i_{fp} are the number of true positives and the false positives of camera C_i, respectively. By varying the detection threshold from the minimum to the maximum value, we can obtain the overall ROC curve.

2.6 Challenges

2.6.1 Reference Images Corrupted by Scene Details

Most of the current methods assume that the contaminations from scene details only exist in the query images while the reference images are relatively clean. However, this assumption may not hold in real-world scenarios as this contamination may occur in both query and reference images. Table 2.1 summarises two common cases a source camera identification system may encounter in the real-world applications. Most of the current methods consider the first case that the reference SPN is uncontaminated while the second case has not yet received much attention. To address this problem, we will introduce our proposed algorithm which is able to deal with contaminations in the reference set in the next chapter.

2.6.2 Source Camera Identification in Large Databases

The methods proposed in different modules can be combined together to achieve a very high accuracy. However, when a large database is concerned, the SPN-based identification system presents its unique set of challenges, which revolve around two main issues. The first issue relates to the high dimensionality of SPN. As a result, main memory operations like loading of SPN takes considerable amount of time. At the same time, each SPN needs a fairly large amount of space for storage. Furthermore, since the SPN looks more like random signal, compression is not very effective. Typically, the SPN extracted from a 10 megapixel image may take up to 50 MB of space even after compression. The second issue is the computational complexity of the matching algorithm. The matching process involves vector operations which, when combined with the high dimensionality of data, becomes a critical concern.

In order to address the issue of prohibitively computational complexity caused by the high dimensionality of SPN, many efforts have been made in recent years. The proposed methods in the literature can be divided into two categories. The methods of the first category attempt to reduce the number of correlations so that there is a smaller number of multiplications to be done. In [62, 63], Bayram *et al.* proposed to organize a database of reference SPNs into a binary search tree. In such a binary search tree, each leaf node represents a reference SPN from the database. Each internal node is the composite SPN which is composited from all the reference SPNs at the leaf nodes in the subtree beneath it. This composite SPN is defined as the normalized sum of all the reference SPNs beneath it. By organizing all the reference SPNs in such a tree, it allows matching multiple reference SPNs in a single verification process. For example, if a query SPN looks for a matched reference

SPN in a binary search tree, it traverses the tree from root to leaf, matches with the SPN at each node of the tree and makes decision. If the decisionss is positive, the searching is then continued in the subtree beneath it. But if the decision is negative, then no more comparisons are needed. On average, this means that each comparison allows the operations to skip about half of the rest tree so that each retrieval takes time proportional to the logarithm of the number of the reference SPNs stored in the tree. Compared to the method based on linear search, this method is more computationally effective. However, there is a trade off between efficiency and accuracy. Since the probability of error increases with the number of reference SPNs in a tree, this method is much less accurate than the method based on the linear search, especially when a large number of reference SPNs are stored in a single binary tree. Thus, it requires to construct multiple binary search trees so as to maintain a desirable identification accuracy. As a result, this method eventually requires to calculate a $(L/t)\log t$ number of correlations, where L is the number of all the reference SPNs and t is the number of the reference SPNs in each tree.

Approaches of the second category aim to lower the computational complexity by compressing the large-sized SPN. In [64–66], the authors introduced a fingerprint digest as a possible solution. The authors assume that the larger components (in magnitude) of a reference SPN is more reliable than the small ones and thus should be used in correlation detection while the small components can be discarded. Thus, this fingerprint digest is primarily formed by keeping only k elements of a reference SPN with the highest energy values and their locations. As a result, the dimensionality of this fingerprint digest is lower than that of the normal-sized SPN. Since the complexity of calculating the correlation is proportional to the size

of SPN, the method based on fingerprint digest would boost the matching efficiency by N^2/k times, where N^2 is the size of the normal-sized SPN (*e.g.,* $N^2 = 512^2$ and $k = 50000$). An improved search strategy based on fingerprint digest is proposed in [67] and [68].

In [69], Bayram *et al.* proposed to represent an SPN by binarizing the values of each pixel. Essentially, the authors only use the sign information of each element of the query or reference SPN and completely disregard the magnitude information. Therefore, by performing this method, each element of a real-valued SPN would be binarized into either -1 or $+1$. As a result, a binarized SPN only requires 1 bit to store each element, while a real-valued SPN requires 64 bit for each element. Although this method does not reduce the dimensionality of SPN, it can considerably reduce the storage requirements and speed up the time of loading SPN to memory so that indirectly boost fingerprint matching process. However, this method would inevitably cause the degradation in matching accuracy due to loss of information caused by binarization.

Valsesia *et al.* in [70–72] proposed to compress the sensor fingerprint via a random projection. This method is based on the Johnson-Lindenstrauss (JL) lemma [73]. The JL lemma states that a small set of points in a high-dimensional space can be embedded into a space of much lower dimension in such a way that distances between the points are nearly preserved. Based on this concept, a random matrix can be found which satisfies the JL lemma. Then, an N^2-dimensional SPN is projected into an m-dimensional subspace, with $m < N^2$, by using the obtained random matrix. By doing so, the dimensionality of the original SPN can be reduced from N^2 to m. However, the method also causes penalties to the matching accuracy

during compressing the SPN.

As shown above, many efforts have been made to improve the efficiency of source camera identification in recent years. However, the results showed that while these compressing methods bring significant computation reduction, they also undesirably decrease the identification accuracy in the meantime. In the light of this limitation, we will introduce our algorithms which aims at improving the computational complexity of source camera identification without degrading the identification accuracy in the following chapters.

Chapter 3

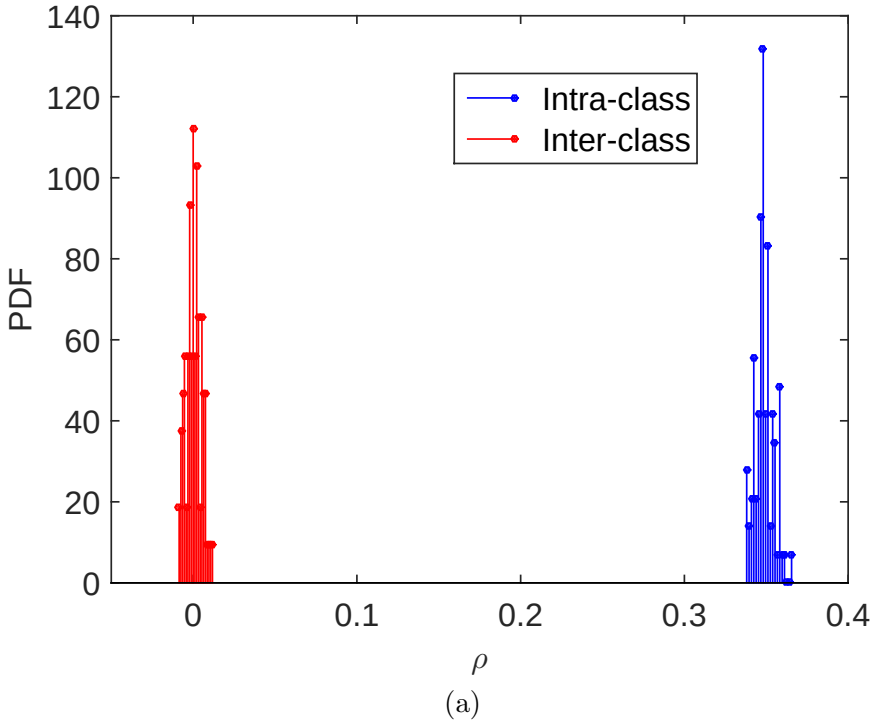
Reducing the Impact of Scene Details in Source Camera Verification

3.1 Problem Statement

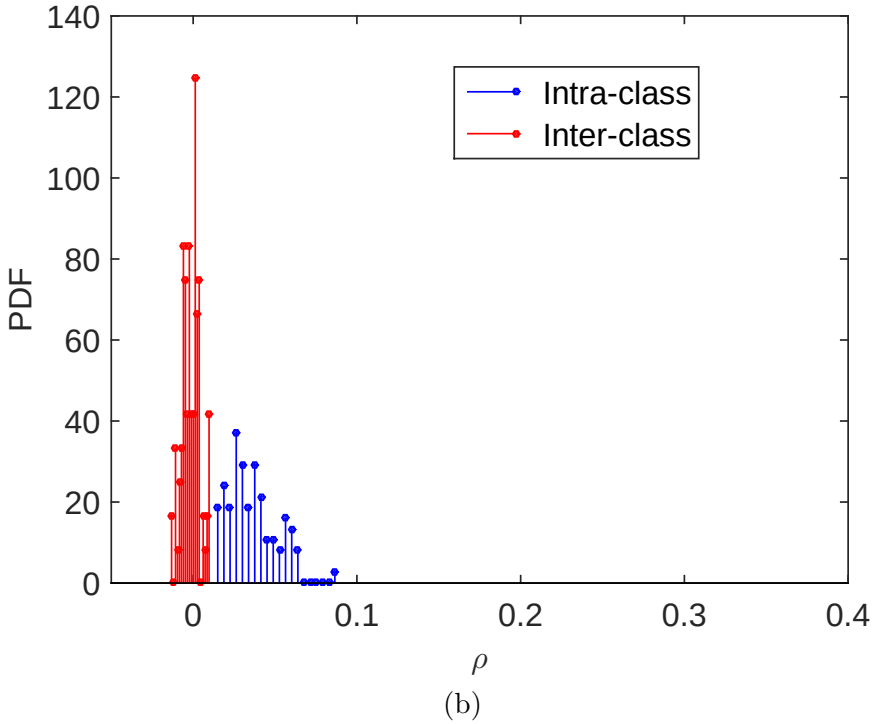
Typically, the reference SPN of a camera is normally estimated by averaging multiple clean SPNs extracted from the blue sky images taken by that camera. However, in practical environments investigators may only have some natural images with strong scene details rather than blue sky images for the reference SPN estimation. As mentioned before, SPN can be severely contaminated by scene details. Thus, with such contaminations in the reference SPN, it may lead to a false identification. For example, Fig. 3.1 shows the estimated inter-class (in red *colour*) and intra-class (in blue colour) Probability Density Functions (PDFs) of the correlation coefficients

ρ between query and reference SPNs. When both query and reference images are blue sky images (Fig. 3.1 (a)), the intra-class and inter-class distributions are well separated so that it is easy for investigators to set a decision threshold and achieve an accurate identification. However, when both query and reference images contain significant scene details (Fig. 3.1 (b)), the intra-class and inter-class distributions are almost mixed together, making it more difficult to draw an accurate decision. Although there have been several studies dedicated to improving the performance of the SPN-based source camera verification, the problem of estimating trustworthy reference SPN from images with strong scene details is still less studied. In order to solve this problem, a context adaptive reference SPN estimator is proposed in this chapter to further improve the performance of source camera verification. Moreover, we consider the situation that the number of reference images of the questioned camera is limited (i.e., 15 reference images per camera), which is a case most current works do not take into account. Experimental results show that the proposed method achieves better performance than the schemes based on the averaged reference SPN, especially when only a few reference images are used.

The rest of this chapter is organized as follows. In the next section, we first introduce a metric which can measure the quality of the SPNs extracted from different image regions. In Section 3.2, the proposed reference estimator is introduced in details. Experimental results are reported and discussed in Section 3.3. Finally, the conclusion is set out in Section 3.4.



(a)

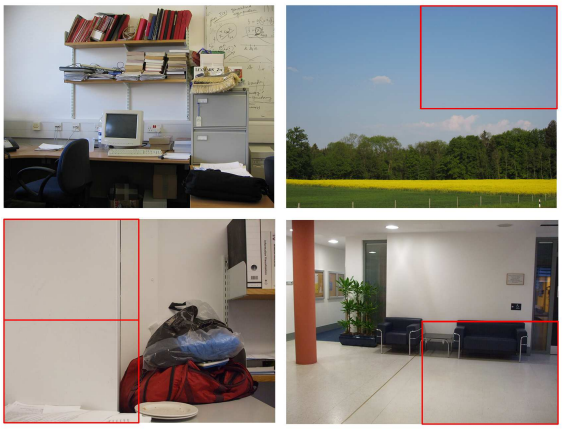


(b)

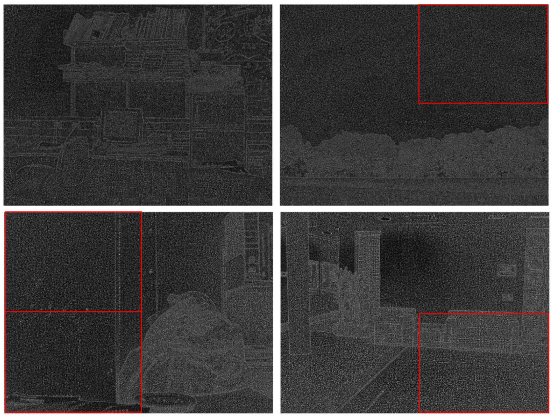
Figure 3.1: Intra-class and inter-class PDFs of the normalized cross-correlation value calculated from SPNs extracted from different kinds of images (a) low-variation images, such as blue sky and flat wall images (b) Images containing scene textures.

3.2 Context Adaptive Reference SPN Estimator

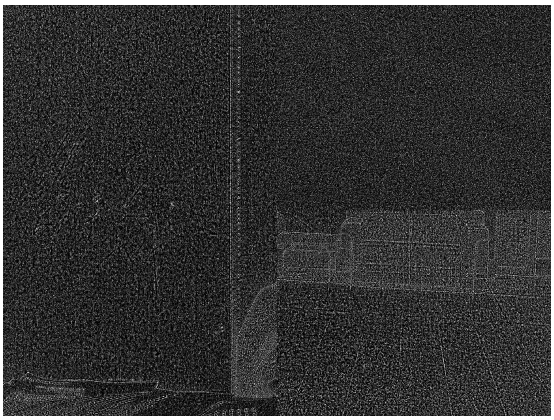
Although the SPN extracted from images with strong scene details can be severely contaminated, it is possible that not all the regions within an image contain strong scene details. As shown in Fig. 3.2 (a), images with strong scene details may still have some smooth regions that just like the blue sky or flat wall. Needless to say, such smooth regions contain SPNs with better quality than the regions with strong scene details. Considering the variety of SPN quality within any ordinary images with scene details, we therefore propose to construct a more reliable reference SPN estimator. A simple example is presented in Fig. 3.2 to briefly explain this idea. Fig. 3.2 (a) shows four images with strong scene details, while there are still some smooth regions within each of these images. Fig. 3.2 (b) presents the SPNs extracted from the image Fig. 3.2 (a). In order to estimate a reliable SPN from these 4 images, first of all we roughly divide each image into 4 blocks. Therefore, for each location, e.g., left-top corner, we would have 4 image blocks from these 4 images. In terms of smoothness, we mark out the optimal image block in red among all the image blocks from the same location. Then we collect the SPN from these optimal image blocks to form a composite SPN, i.e., Fig. 3.2 (c). As shown in Fig. 3.2(b), the SPNs extracted from these optimal blocks contain fewer scene details than the SPNs extracted from other image blocks. Thus, the composite SPN (as presented in Fig. 3.2(c)), which is made up by the optimal SPN blocks from different locations, contains fewer scene details than any original SPNs in Fig. 3.2(b). In the same manner, we can generate another 3 composite SPNs by using the rest SPN blocks left in Fig. 3.2 (b). It is not difficult to deduce that the quality



(a)



(b)



(c)

Figure 3.2: (a) Four natural images with scene details. All the four images are taken by the same camera. (b) Four SPNs extracted from the images in (a). (c) A composite SPN consists of blocks collected from the SPNs in (b).

of these 3 new generated SPNs would be worse than the that of the one present in Fig. 3.2(c). Considering that the conventional averaging method treats all the SPN blocks equally, we therefore can obtain a more reliable reference SPN by assigning higher weights to the composite SPNs with SPN blocks collected from smoother image regions.

3.2.1 SPN Quality Measurement

To achieve the aforementioned idea, we first introduce a measurement to evaluate the quality of the SPNs extracted from different image blocks. As mentioned in [7], there are two main factors that are most crucial in determining the quality of the SPN in an image block.

The first factor is the amount of scene details contained in an image block. Generally speaking, an image block that contains fewer scene details would be a smoother region and thus can provide a cleaner SPN. This concept has been validated in Fig. 3.2. In this work, we use the image entropy as the measurement to describe the amount of scene details in an image. Entropy is a statistical measure of randomness that can be used to characterize the degree of details in images [74]. The image with high entropy value would have a great deal of contrast from one pixel to the next, such as images of heavily cratered areas on the moon, while the images with low entropy value would have very little contrast and large number of pixels with the same or similar intensity values, such as those containing a lot of blue sky or flat wall [75]. An image that is perfectly flat would have an entropy of zero. Therefore, the image block with lower entropy value would contain SPN with

better quality. The entropy of an image block is calculated as follows

$$E(\mathbf{I^b}) = -\sum_{k=0}^{255} p_k \log_2 p_k, \tag{3.1}$$

where k is a gray level of a pixel, p_k is the probability of the gray level k in the image region $\mathbf{I_b}$.

The other determinative factor is the image brightness/luminance. As mentioned in Chapter 2, the discriminative part of SPN, i.e., PRNU, is a multiplicative noise. From the Eq. (2.2), the extracted noise residual $\mathbf{X}$ can be simply modelled as

$$\mathbf{X} = \mathbf{IK} + \mathbf{\Xi}, \tag{3.2}$$

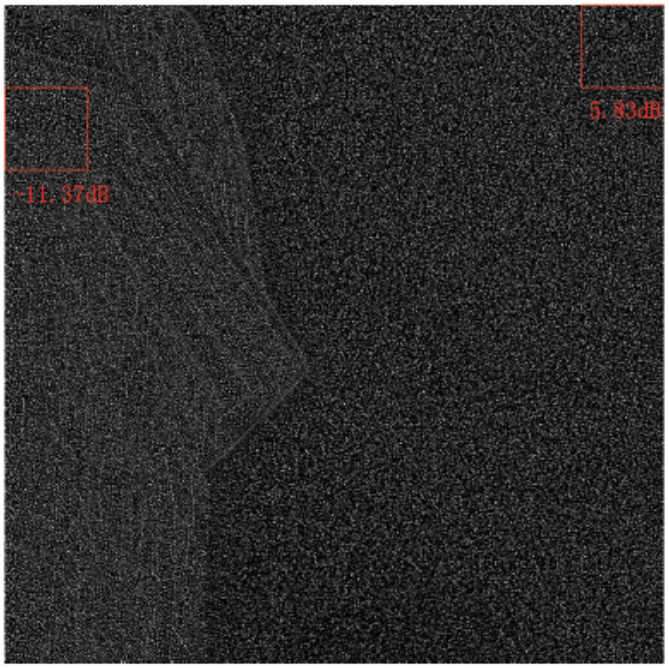
where $\mathbf{IK}$ presents the true SPN term and $\mathbf{\Xi}$ is a mixture of noises. From Eq. (3.2), we can see that the amplitude of the true SPN $\mathbf{IK}$ is proportional to the luminance of its source image $\mathbf{I}$. Thus, the brighter the image $\mathbf{I}$ is, the stronger the true SPN term $\mathbf{IK}$ would be. Under this circumstance, strengthening the true SPN $\mathbf{IK}$ would relatively lower the impact of the additive noise term $\mathbf{\Xi}$. Therefore, the image block with higher brightness would contain stronger SPN. It is worth mentioning that the brightness of $\mathbf{I}$ should be as high as possible but not saturated because saturated pixels are filled to their full capacity (i.e., $\mathbf{I}[i] \approx 255$) and thereby carry no information about the SPN signal. It is also clear that the pixels of very dark regions would have very low magnitude (i.e., $\mathbf{I}[i] \approx 0$) so that the SPN signal from dark regions would be very weak (i.e., $\mathbf{IK}[i] \approx 0$). Therefore, the SPNs extracted from dark and saturated regions should not be taken into account for the reference SPN estimation.

Considering both of the two determinative factors mentioned above, we propose a measurement Q to evaluate the quality of the SPN extracted from different image blocks, which is modelled as

$$Q(\mathbf{I^b}) = \begin{cases} \frac{B(\mathbf{I^b})}{E(\mathbf{I^b})}, & if \quad 10 < B(\mathbf{I^b}) < 245 \\ 0, & otherwise \end{cases} \tag{3.3}$$

where $B(\mathbf{I^b})$ is the average brightness of image block $\mathbf{I^b}$, $E(\mathbf{I^b})$ is the entropy of $\mathbf{I^b}$. According to Eq. (3.3), a smooth and bright image block would be assigned with a high quality score. As mentioned above, the SPNs extracted from dark and saturated regions are very weak. Therefore, according to our experiments, the image block $\mathbf{I^b}$, with $B(\mathbf{I^b}) \leqslant 10$ or $B(\mathbf{I^b}) \geqslant 245$, is assigned with the lowest quality score, i.e., $Q(\mathbf{I^b}) = 0$. As shown in Fig. 3.3, an example is used to validate the feasibility of the proposed measurement. Fig. 3.3 (a) shows an image taken by a Canon Ixus70, and Fig. 3.3 (b) is the SPN extracted from Fig. 3.3 (a) by using the Basic method [6]. There are two image blocks (marked by red block) in Fig. 3.3 (a). The one on the right side is a smooth region with high luminance, and its SPN quality score Q is 22.81. The left one is a region with strong scene details, thus it has a relatively low quality score, i.e., 8.56. Since we have the source camera, we can estimate the ground truth (i.e., the true SPN) for these two image blocks. By subtracting the ground truth from the extracted SPNs, we can approximately estimate the contaminants left in these two SPN blocks. Thus, the signal-to-noise ratio (SNR) for these two SPN blocks can be calculated. As indicated by the obtained quality score, the smooth image block has a higher average SNR, i.e., 5.83dB, which is much higher than the one with scene details, i.e., -11.37dB.

(a)



(b)

Figure 3.3: (a) An image taken by Canon Ixus70. (b) The noise residual extracted from (a) using the Basic method. Note the intensity of (b) has been downscaled 3 times for visualization purpose.

3.2.2 Methodology

Based on the measurement introduced in the last section, the proposed reference SPN estimator can be built in the following five steps. The details for each step are described as follows:

1. **SPN extraction.** Assume there are a set of images $\{\mathbf{I}_i\}_{i=1}^{N}$ taken by the same camera. In the case of colour images, we perform the SPN extraction on each colour channel. For each $\mathbf{I}_i$, we extract the SPNs $\mathbf{X}_i^R$, $\mathbf{X}_i^G$ and $\mathbf{X}_i^B$ in spatial domain from the red, green and blue channel respectively by using one of the SPN extraction method mentioned in Section 2.2, such as the Basic method [6]

$$\mathbf{X}_i^{(R,G,B)} = DWT^{-1}\{DWT(\mathbf{I}_i^{(R,G,B)}) - F[DWT(\mathbf{I}_i^{(R,G,B)})]\}, \tag{3.4}$$

   where DWT is discrete wavelet transform and F indicates the Michat filter [40]. The obtained $\mathbf{X}_i^R$, $\mathbf{X}_i^G$ and $\mathbf{X}_i^B$ are then combined into a grayscale SPN $\mathbf{X}_i$ by applying the common RGB-to-gray conversion, such as

$$\mathbf{X}_i = 0.299\mathbf{X}_i^R + 0.587\mathbf{X}_i^G + 0.114\mathbf{X}_i^B. \tag{3.5}$$

2. **Segmentation.** In this step, we first divide each full-sized image $\mathbf{I}_i$ and SPN $\mathbf{X}_i$ into M non-overlapping image blocks $\{\mathbf{I^b}_{ij}\}_{j=1}^{M}$ and SPN blocks $\{\mathbf{X^b}_{ij}\}_{j=1}^{M}$ in the same manner. Thus, each image block $\mathbf{I^b}_{ij}$ is associated with its SPN block $\mathbf{X^b}_{ij}$, where j is their location label. The size of each image block and SPN block is S.

3. **SPN quality evaluation and ranking.** We evaluate the SPN quality score

Q_{ij} for each image block $\mathbf{I^b}_{ij}$ by using the Eq. (3.1). Then the obtained quality score Q_{ij} is assigned to the corresponding SPN block $\mathbf{X^b}_{ij}$. For each location j, the SPN blocks $\{\mathbf{X^b}_{ij}\}_{i=1}^{N}$ from N images are sorted in the descending order according to their quality score Q_{ij}. By doing so, the SPN block with a higher quality score would have a higher ranking value (with 1 as the highest).

4. **SPN composition.** The SPN blocks from different locations with the same ranking value are combined to form a composite SPN. By repeating this process N times, N composite SPNs $\{\hat{\mathbf{X}}_i\}_{i=1}^{N}$ can be generated, each with the ranking value i. By doing so, we would generate not only the composite SPNs containing clean SPN blocks, but also the composite SPNs which are full of scene details (e.g., the composite SPNs with lower ranking value). These low-quality SPNs are also taken into account for the reference SPN estimation because they can still contribute on suppressing random noise, especially when the number of available reference images is inadequate.

5. **Weighting.** Instead of treating all SPNs equally, we assign higher weight to the composite SPN with higher ranking value. The weighting factor ω_i for the composite SPN $\hat{\mathbf{X}}_i$ is calculated as follows

$$\omega_i = \frac{2(N+1-i)}{N(N+1)}, i \in [1, N]. \tag{3.6}$$

Finally, the reference SPN $\mathbf{R}$ is estimated as

$$\mathbf{R} = \sum_{i}^{N} \omega_i \hat{\mathbf{X}}_i, i \in [1, N]. \tag{3.7}$$

It is worth noting that different settings of block size S might affect the estimation of the reference SPN $\mathbf{R}$. Intuitively, applying a smaller S is more likely to achieve a more accurate estimation. The reason is that the quality score Q depicts the average SPN quality of an image block, thus Q would reflect the SPN quality of an image block more accurately when the block size becomes smaller. However, when the block size is too small, the computational cost would be exorbitant. More details about the settings of S would be discussed in Section 3.3.2.

3.3 Experiments and Discussion

3.3.1 Experimental Setup

In this work, 16 cameras from the Dresden Image database [76] were chosen and used in our experiments. A total of 2400 images from 16 cameras are involved in this experiment, each camera is responsible for 150 image. These 16 cameras belong to 4 camera models and each camera model has 3 or 5 devices. The information of these cameras are listed in Table 3.1. All images are natural pictures with strong scene details, which were taken under a wide variety of natural indoor and outdoor scenery. For each camera, we randomly separate 150 images into two sub-image datasets, namely the reference dataset and test dataset, with 30 and 120 images, respectively. The images in the reference dataset are used for reference SPN estimation while images in the test dataset are used as test samples. To avoid the vignetting effects [77], all the experiments are performed on the image blocks of three fixed sizes cropped from the center of the full size images, which are 128×128, 256×256 and 512×512 pixels respectively. We extract noise residuals from three colour (i.e., red,

Table 3.1: 16 Cameras involved in our experiments.

Cameras	**Resolution**	**Cameras**	**Resolution**
Canon_Ixus70_A	3072×2304	Samsung_L74wide_A	3072×2304
Canon_Ixus70_B	3072×2304	Samsung_L74wide_B	3072×2304
Canon_Ixus70_C	3072×2304	Samsung_L74wide_C	3072×2304
Nikon_CoolPixS710_A	4352×3264	Olympus_mju_1050SW_A	4352×3264
Nikon_CoolPixS710_B	4352×3264	Olympus_mju_1050SW_B	4352×3264
Nikon_CoolPixS710_C	4352×3264	Olympus_mju_1050SW_C	4352×3264
Nikon_CoolPixS710_D	4352×3264	Olympus_mju_1050SW_D	4352×3264
Nikon_CoolPixS710_E	4352×3264	Olympus_mju_1050SW_E	4352×3264

green and blue) channels and combine them by using Eq. (3.5).

3.3.2 Parameter Settings and Discussion

The proposed method has only one parameter that need to be set, namely, the size of image and SPN block S. As mentioned in Section 3.2.2, different settings of S might affect the estimation of reference SPN. On the one hand, applying a smaller size would make the SPN estimator more adaptive to scene textures so that it might improve the accuracy of reference SPN estimation. On the other hand, when a small block size (e.g., 16×16 pixels) is applied, the computational complexity of the proposed method can be prohibitive. Intuitively, there is a trade-off between accuracy and efficiency. In order to find a proper setting of S, the proposed method is tested on three different SPN blocks sizes (i.e., $S = 16 \times 16$, 32×32 and 64×64 pixels). From Table. 3.2, we can see that the TPR of the proposed method performed on the small image block (i.e., $S = 16 \times 16$ pixels) is just slightly larger than that on the large image blocks. It indicates the performance of the proposed method is not very sensitive to different settings of S. Considering

Table 3.2: The TPR (with the FPR fixed at 10^{-3}) of different methods with respect to different settings of S.

Method	**The size of image/SPN block (pixels)**		
	16×16	32×32	64×64
Basic+Proposed	0.346	0.333	0.327
BM3D+Proposed	0.425	0.413	0.405

the fact that using a small block size would cause more computational cost, therefore we recommend setting $S = 64 \times 64$ pixels as it is 2 and 4 times faster than using $S = 32 \times 32$ and $S = 16 \times 16$ pixels.

3.3.3 Performance Evaluation

In this work, the overall ROC curve [13] is used to present the performance of the proposed method. In order to validate the feasibility of the proposed method, we compare it with the traditional averaging method in conjunction with two different SPN extraction methods, i.e., Basic [6] and BM3D [8]. In this work, we consider the situation that the number of reference images per camera is inadequate (i.e. $N < 50$), which is a case most current works do not take into account. Therefore, we estimate the reference SPN for each camera by only using $N = 15$ and $N = 30$ images from the reference dataset. For each camera, the SPNs extracted from the 120 testing images of this camera are used as the positive samples and the SPNs obtained from the rest 1800 testing images of the other 15 cameras are deemed as the negative samples. Therefore, we would have 120×16 positive and 1800×16 negative samples from all the 16 cameras in total. To get convincing results, all these positive and negative samples are used together to draw the overall ROC curve.

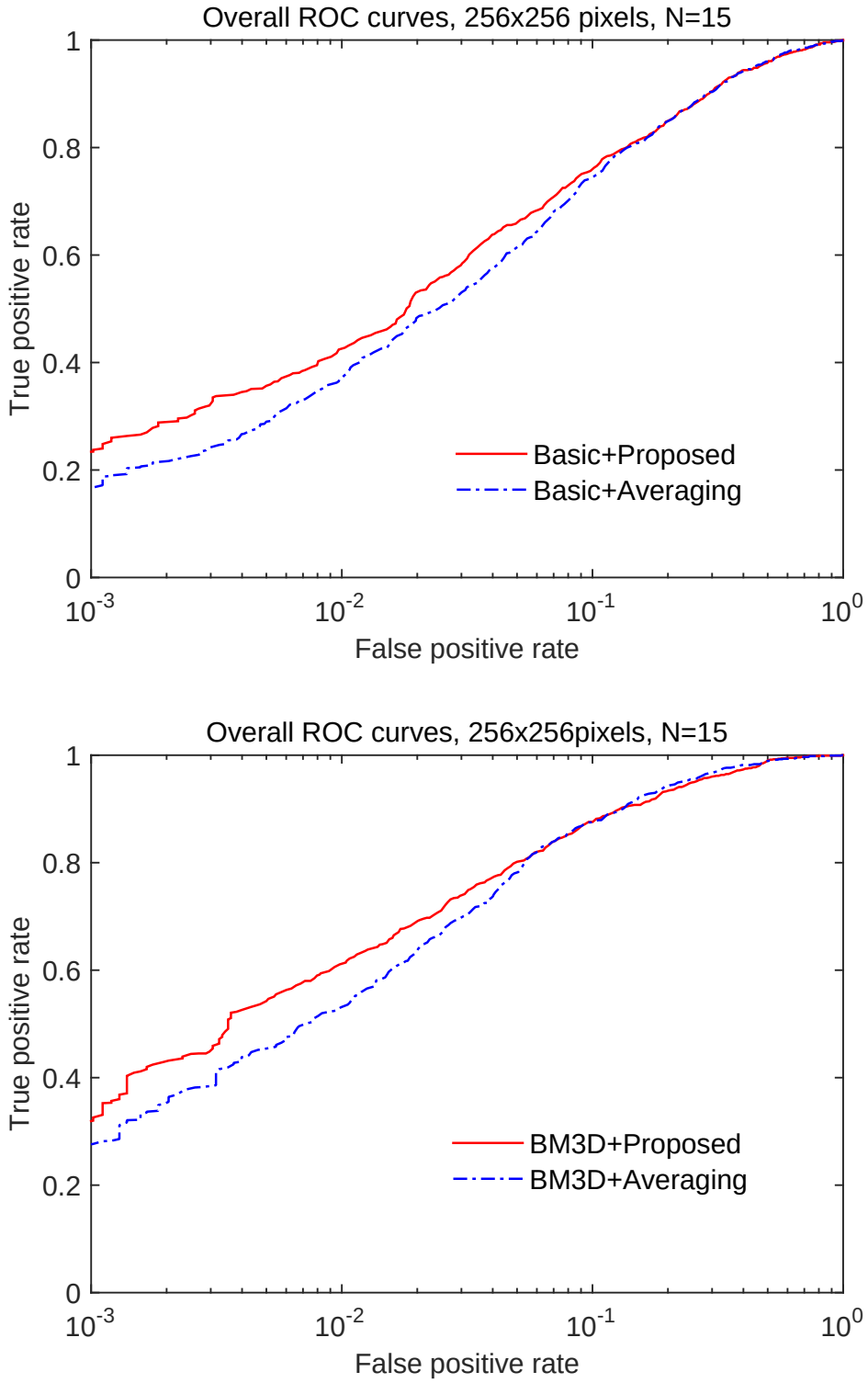


Figure 3.4: The overall ROC curves of difference methods with 15 reference images based on images with size of 256×256 pixels.

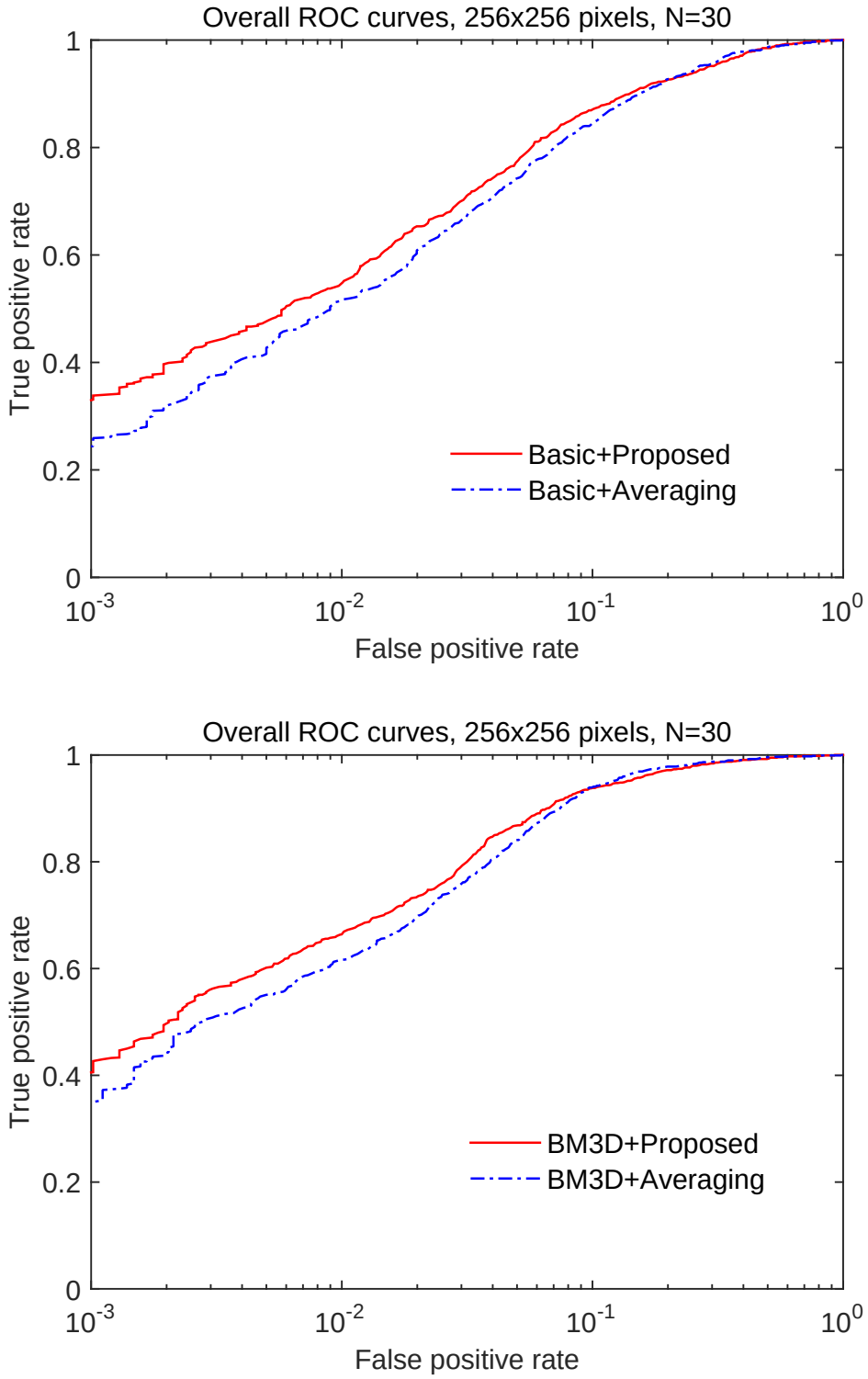


Figure 3.5: The overall ROC curves of difference methods with 30 reference images based on images with size of 256×256 pixels.

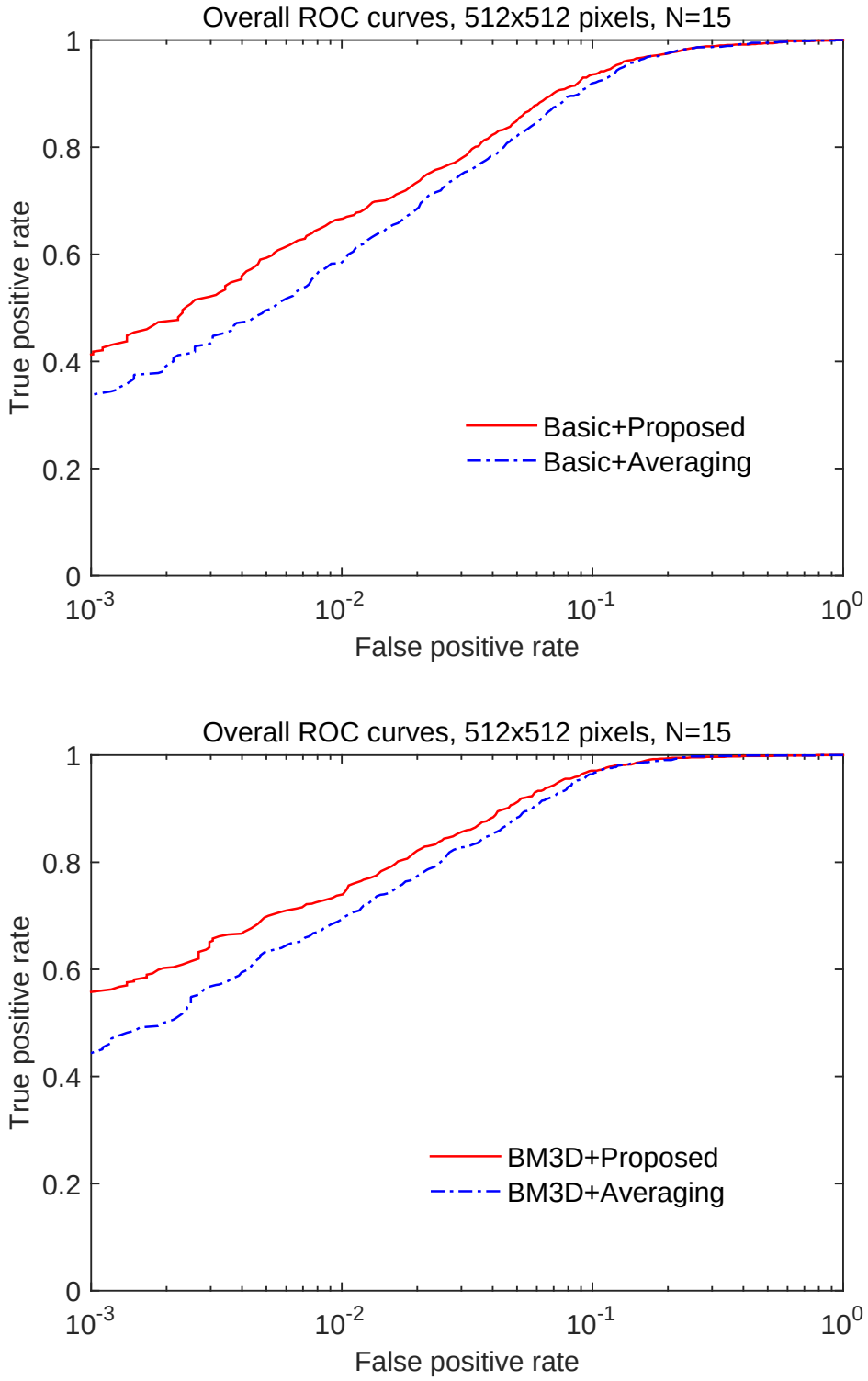


Figure 3.6: The overall ROC curves of difference methods with 15 reference images based on images with size of 512×512 pixels.

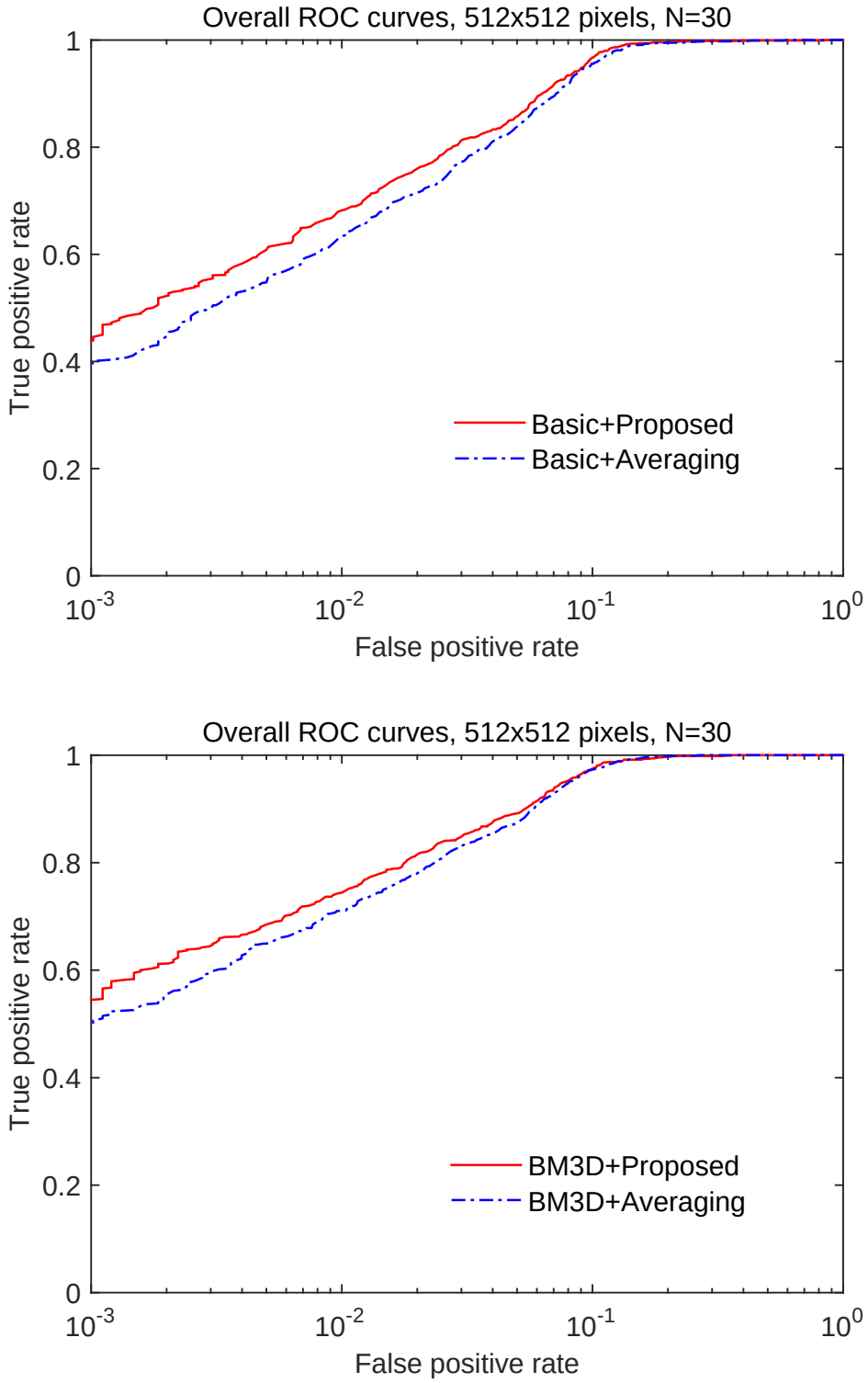


Figure 3.7: The overall ROC curves of difference methods with 30 reference images based on images with size of 512×512 pixels.

The overall ROC performance of different methods with respect to different image sizes and different numbers of reference images are shown in Fig. 3.4 - Fig. 3.7. In this experiment, Basic/BM3D+Proposed indicates that SPNs are extracted by using Basic/BM3D, and reference SPN is estimated by using the proposed SPN estimator; and Basic/BM3D+Averaging means that reference SPN is estimated by the traditional averaging method. In the real-world applications, a low false positive rate is usually required so as to ensure a low probability of wrong accusation. Therefore, in order to show the details of the ROC curves with a low FPR, the horizontal axis of all the overall ROC curves are plotted in the logarithmic scale. As shown in Fig. 3.4 - Fig. 3.7, the proposed method (red curves) outperforms the traditional averaging method regardless of the image size, the SPN extraction method and the number of reference images. It indicates that the proposed method is more reliable than the traditional averaging method on estimating reference SPN from a noisy image. Moreover, by comparing Fig. 3.6 with Fig. 3.7, we can see that the proposed method is more superior to the averaging method when the number of reference images is small (i.e., $N = 15$). These observations suggest that the proposed method can bring additional performance gains to a verification system when reference images are contaminated by scene details and the number of reference images is limited. It is worth mentioning that the BM3D method consistently outperforms the Basic method on the overall ROC performance. As mentioned in Section 2.2.1, this is because the BM3D method is superior to the Basic method on suppressing the impact of scene details.

Table 3.3 shows the TPR of different methods at a low FPR of 10^{-3}. Similar to the observation in the overall ROC curve analysis, the proposed method

Table 3.3: The TPR (with the FPR fixed at 10^{-3}) of different methods with respect to different number of reference images on different image sizes.

Method	256×256		512×512	
	15	30	15	30
Basic+Averaging	0.165	0.244	0.337	0.401
Basic+Proposed	0.234	0.329	0.416	0.439
BM3D+Averaging	0.276	0.348	0.442	0.498
BM3D+Proposed	0.319	0.405	0.555	0.549

consistently achieves higher TPR than the averaging method under all conditions. Moreover, we can see that when the number of reference images decreases from 30 to 15, the TPR of proposed method drops more slowly than the averaging method especially on the large image size (i.e., 512×512 pixels). For example, on the 512×512 pixels images, the TPR of Basic+Averaging degrades by 15.96% (from 0.401 at $N = 30$ to 0.337 at $N = 15$), while the TPR of Basic+Proposed only drops by 5.23% (from 0.439 at $N = 30$ to 0.416 at $N = 15$). More specifically, for B-M3D+Proposed on the 512×512 pixels images, the TPR at $N = 15$ is even slightly larger than that at $N = 15$. It implies that the proposed method is more reliable than the traditional averaging method when the number of the available reference images is limited.

3.4 Conclusion

In this chapter, we introduced a measurement based on the local image entropy and luminance to evaluate the SPN quality of different image blocks for the reference SPN estimation. Based on this measurement, a novel reference SPN estimator is proposed to improve the performance of source camera verification. By weighting

the SPN blocks according to their corresponding quality, the proposed estimator can estimate a more reliable reference SPN from the limited number of images with scene details. Experimental results show that our method achieves better results than the traditional averaging method, especially when there are only few reference images ($N = 15$) available. These results suggest that the proposed reference SPN estimation method is more practical for solving the problem of source camera verification with the reference images contaminated by scene details.

Chapter 4

A Compact Representation of Sensor Pattern Noise

4.1 Problem Statement

In literature, many efforts have been devoted in the SPN-based SCI system, which allows the current SCI system almost achieve a perfect identification accuracy. However, the SPN-based SCI system presents its unique set of issues, which relate to the high dimensionality of SPN. The dimensionality of SPN is as great 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 SPN. Thus with a large number of reference SPNs in the database to be matched, the complexity of the matching step would become a critical concern.

In order to address the high complexity issue, many efforts have been made in recent years. The methods in literature can be roughly divided into two categories.

The methods of the first category attempt to reduce the number of correlations so that there is a smaller number of multiplications to be done. In [62, 63], Bayram et al. proposed to organize a database of reference SPNs into a binary search tree. Each internal node in a binary tree is represented by an SPN composited from all the SPNs at the leaf nodes in the subtree beneath it. By applying this method, the total number of correlations to be calculated is significantly reduced.

The approaches of the second category aim to lower the computational complexity by compressing SPN. In [64–66], the authors introduced a fingerprint digest as a possible solution. This fingerprint digest is primarily formed by keeping only the k elements of the fingerprint with the highest energy values. This results in N^2/k times reduction in terms of computation time. In [69], Bayram et al. proposed to represent an SPN by binarizing the values of each pixel. Valsesia et al. in [70, 71] proposed to compress the sensor fingerprint via a random projection.

However, while these compression methods can bring significant computation reduction, they also undesirably decrease the identification accuracy. In the light of this limitation, in this chapter, we aim to improve the computational efficiency of SCI without degrading the identification accuracy. We employ the concept of PCA denoising [26–28, 31] in the task of SCI. An effective feature extraction algorithm based on this concept is proposed to extract a small set of components from the original noise residual, which tends to carry most of the information of the true SPN signal. While this algorithm is based on the assumption that the training set is well representative of the population so that an effective 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 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 give more details and discussion of the 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 SPN 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 chapter is organized as follows. In Section 4.2, we present the proposed training dataset construction procedure and the feature extraction method in details. Then, the proposed source camera identification method is summarized. Extensive experimental evaluations are provided in Section 4.3. Section 4.4 concludes.

4.2 PCA-based Feature Extraction Algorithm

Generally speaking, the high-dimensional data would not only incur a costly computation but also tend to contain more redundancy and interfering components. For simplicity, we write Eq. (2.2) as the sum of the true SPN term and unwanted noises, given by

$$\mathbf{X} = \mathbf{X}^{(0)} + \mathbf{\Xi}, \tag{4.1}$$

where $\mathbf{X}^{(0)}$ is the SPN term, $\mathbf{\Xi}$ is an additive mixture of the unwanted interferences, which may include image scene details noises (referred to as contaminated images), and the artifacts introduced by color interpolation, JPEG compression,

camera processing operations [7], etc. The former can be image-scene-specific, while the latter can be shared among cameras of the same model or sensor design, so they are non-unique, less discriminant and redundant. To improve the performance of SCI systems, one intuitive way is to suppress these artifacts Ξ.

PCA [78] is a well-known unsupervised learning method, which minimizes the reconstruction error using a linear transformation, can be used to learn compact representation. This method has been widely used for the purposes of denoising [26, 28], dimensionality reduction [79], feature extraction [80], etc. Compared with the data-independent dimensionality reduction method random projection, PCA projection matrix has to be learned based on a training data, and it generally has higher performance in classification tasks [81]. In this work, we attempt to find a PCA transformed domain where the true SPN signal is well represented. Ideally, by projecting the extracted noise residuals into this domain, a small set of coefficients that contain most of the representative information of the true SPN signal can be extracted.

4.2.1 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 represent 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 represented 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. 4.1(b)). Without removing these strong contamination 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 unwanted noises in the training set:

1. **Training sample selection.** To build the training set, if we have the access to the cameras in database, we give the priority to the noise residuals extracted from flatfield 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. 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 the cameras to take flat-field images. Instead only some images with varying scene details taken by those camera are available (e.g., from someone's Facebook account). In this case, our strategy is to suppress the impact of scene details by averaging. Considering the fact that scene details presented in different images are normally different, we can generate a more 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. (4.1). In this context $\boldsymbol{\theta}$ represents the scene details noises, 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 suppressed by 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 $\sigma^2_{\mathbf{X}} = \mathbb{E}[(\boldsymbol{\theta}_i - \frac{1}{N}\sum_j^N \boldsymbol{\theta}_j)^2], \quad i = 1, 2, ..., N$, respectively. For a camera if we average the SPNs of a random subset

of T out of the N reference images for L times, then according to Eq. (4.1) we will have

$$\mathbf{X}'_l = \hat{\mathbf{X}} + \frac{1}{T}\sum_{t}^{T} \boldsymbol{\theta}_{lt}, \quad l = 1, 2, ..., L. \tag{4.2}$$

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

$$\mu_{\mathbf{X}'} = \hat{\mathbf{X}} + \frac{1}{LT}\sum_{l}^{L}\sum_{t}^{T} \boldsymbol{\theta}_{lt}, \tag{4.3}$$

$$\sigma^2_{\mathbf{X}'} = \frac{1}{L}\sum_{l}^{L}(\frac{1}{T}\sum_{t}^{T} \boldsymbol{\theta}_{lt} - \frac{1}{LT}\sum_{l}^{L}\sum_{t}^{T} \boldsymbol{\theta}_{lt})^2. \tag{4.4}$$

In Eq. (4.4), 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. (4.4) also converges to the mean of the scene details $\bar{\boldsymbol{\theta}} = \frac{1}{N}\sum_{i}^{N} \boldsymbol{\theta}_i$, which makes $\sigma^2_{\mathbf{X}'} \to 0$, hence suppressing the interference of scene details.

2. **Training sample enhancement.** Besides scene details, some non-unique artifacts such as CFA pattern and JPEG blockiness may also lead to an inaccurate training. Since these artifacts in the images taken by the cameras of same model or brand are similar (with small variance), they would survive from the averaging operation. Nevertheless, as shown in [54], these artifacts cause peaks in the DFT magnitude spectrum, while the SPN signal 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 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 strategies above, we summarize the proposed training set construction as follows:

1) Extract 2D noise residuals $\{\{\mathbf{X}_{ij}\}_{i=1}^{N}\}_{j=1}^{c}$ from the $W \times W$ center blocks cropped from these n reference images.

2) For each camera C_j, randomly select T images of noise residuals from $\{\mathbf{X}_{ij}\}_{i=1}^{N}$ for averaging.

3) Detect and suppress the peaks of the averaged noise residual in the DFT magnitude specturm via SEA [54]. 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 Step 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 blockiness 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 consequentially 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 may consequently increase the effect of enhancement.

More details about how the setting of T affects performance is discussed in Section 4.3.2.

4.2.2 Feature Extraction in the PCA Domain

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

$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T = \mathbf{A}\mathbf{A}^T, \tag{4.5}$$

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 $k = 1, ..., m$. Noting that the dimensionality of SPN vector can be extremely high (e.g., $m > 10^7$). Therefore, directly decomposing $\mathbf{S} \in \mathbb{R}^{m\times m}$ would incur a prohibitively computational cost (with a complexity $O(m^3)$). To make PCA feasible for the high-dimensional SPN, we apply a fast method instead of computing these eigenvectors when $m \gg n$.

Assume $\mathbf{v}_k{}'$ is the unit eigenvector of $\mathbf{A}^T\mathbf{A} \in \mathbb{R}^{n\times n}$ with eigenvalue λ'_k, and then we can obtain $\mathbf{A}^T\mathbf{A}\mathbf{v}_k{}' = \lambda_k{}'\mathbf{v}_k{}'$. Multiplying both sides by $\mathbf{A}$, we have

$$\mathbf{A}\mathbf{A}^T(\mathbf{A}\mathbf{v}_k{}') = \lambda_k{}'(\mathbf{A}\mathbf{v}_k{}'), \tag{4.6}$$

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 the objective $\mathbf{v}_k$ can be obtained via $\mathbf{v}_k = \mathbf{A}\mathbf{v}_k{}'$. As long as $n < m$, computing eigenvectors via this way 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 associated eigenvalues $\lambda_1 \geqslant \lambda_2 \geqslant ...\lambda_n$. Then a transformed domain can be built as $\mathbf{M}_{pca} = [\mathbf{v}_1, ..., \mathbf{v}_n] \in \mathbb{R}^{m \times n}$. After that, we can apply $\mathbf{M}_{pca}$ to noise residual $\mathbf{x}$ (defined in Eq. (5)) through

$$\begin{aligned} \mathbf{y} &= \mathbf{M}_{pca}^T \mathbf{x} = \mathbf{M}_{pca}^T (\mathbf{x}^{(0)} + \mathbf{\Xi}) \\ &= \mathbf{M}_{pca}^T \mathbf{x}^{(0)} + \mathbf{M}_{pca}^T \mathbf{\Xi} = \mathbf{y}^{(0)} + \mathbf{\Xi}_{\mathbf{y}}, \end{aligned} \tag{4.7}$$

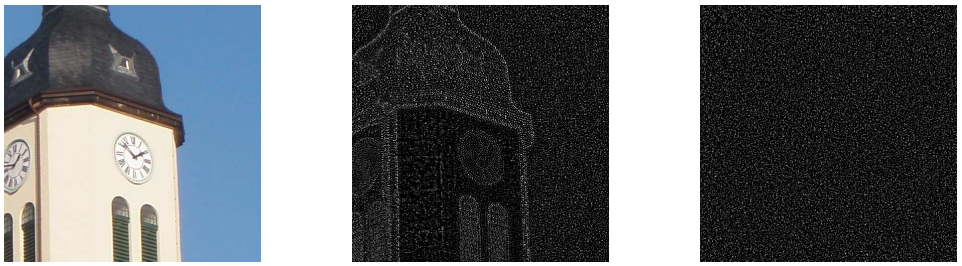
where $\mathbf{y}^{(0)}$ and $\mathbf{\Xi}_{\mathbf{y}}$ are the transformed versions of the SPN term and the noise term, respectively. Now the problem is recast as estimating $\mathbf{y}^{(0)}$ from the noisy $\mathbf{y}$. Generally speaking, in a PCA transformed vector (i.e., $\mathbf{y}$), most energy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

$$d = \min\{d' \mid \sum\nolimits_{i=1}^{d'} \lambda_i / \sum\nolimits_{i=1}^{n} \lambda_i > 98\%\}. \tag{4.8}$$

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

$$\begin{aligned} \mathbf{y}^d &= (\mathbf{M}_{pca}^d)^T \mathbf{x} = (\mathbf{M}_{pca}^d)^T \mathbf{x}^{(0)} + (\mathbf{M}_{pca}^d)^T \mathbf{\Xi} \\ &= {\mathbf{y}^{(0)}}^d + \mathbf{\Xi}_{\mathbf{y}}^d, \end{aligned} \tag{4.9}$$

where $\mathbf{y}^d$ is the dimensionality reduced version of $\mathbf{y}$. According to the feature



(a) (b) (c)

Figure 4.1: (a) An image taken by Olympus_mju_1050SW. (b) The noise residual extracted from (a) using the Basic method. (c) The reconstructed version of (b) via the proposed method. Note the intensity of (b) and (c) has been down scaled 5 times and up scaled 2 times, respectively, for visualization purpose.

vector $\mathbf{y}^d$ and feature extractor $\mathbf{M}_{pca}^d$, we can easily obtain a reconstructed SPN in the spatial domain via the inverse PCA transform as follow

$$\mathbf{x}' = (\mathbf{M}_{pca}^d)\mathbf{y}^d, \tag{4.10}$$

where $\mathbf{x}'$ is an approximation of the original $\mathbf{x}$. If our assumption is correct, noise $\mathbf{\Xi}_\mathbf{y}$ should be suppressed by the PCA-based feature extraction. As a consequence the reconstructed $\mathbf{x}'$ should contain less noises and have a higher signal-to-noise ratio (SNR) than the original noise residual $\mathbf{x}$.

To validate this point, we demonstrate the behaviour of our feature extractor with a simple example. As shown in Fig. 4.1(b), the image content appearing in Fig. 4.1(a) propagates through the Wiener filter into the noise residual. While after performing the proposed feature extraction and inverting the PCA transform, the artifacts caused by scene details have been significantly suppressed in the reconstructed SPN, which can be clearly seen in Fig. 4.1 (c). Then, we calculate the SNR of the signal of interest (i.e., the true SPN) respectively to the contamination left in the noise residual (Fig. 4.1(b)) and the contamination in the reconstructed SPN

(Fig. 4.1(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 noises $\boldsymbol{\Xi}$ in the noise residual $\boldsymbol{\Xi}$ in the noise residual and the reconstructed SPN can be approximately estimated by subtracting the true SPN $\hat{\mathbf{x}}$ from the observed data. 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.

4.2.3 Enhanced Feature Extraction in the LDA domain

In the task of SCI, the class label of the images in database are usually known. By taking advantage of this label information, we can further extract a more discriminant feature by using a supervised learning method, i.e., linear discriminant analysis [82, 83]. The purpose of LDA in this work is to build an enhanced feature extractor $\mathbf{M}_{lda}$, which would better separate different classes. This optimal feature extractor 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 $\mathbf{S}_w$

$$\mathbf{M}_{lda} = \arg\max_{\mathbf{J}} \left| \frac{\mathbf{J}^T\mathbf{S}_b\mathbf{J}}{\mathbf{J}^T\mathbf{S}_w\mathbf{J}} \right|, \tag{4.11}$$

where the within-class scatter matrix $\mathbf{S}_w$ 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 i-th sample of class j, $\boldsymbol{\mu}_j$ is the mean of class j, c is 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 mean of all classes.

Algorithm 4.1. PCA-based Source Camera Identification

Symbols:
m: The dimensionality of the normal-sized 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 4.2.1) to generate a set of training samples $\{\{\mathbf{x}_{ij}\}_{i=1}^{L}\}_{j=1}^{c} \in \mathbb{R}^m$.

2. If $m \gg n$ (in most cases), use the fast method mentioned in Section 4.2.2 to estimate the eigenvectors $\{\mathbf{v}_k\}_{k=1}^{n}$ and the eigenvalues $\{\lambda_k\}_{k=1}^{n}$.

3. Preserve the eigenvectors with the first d largest eigenvalues while discarding the rest to build the 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 $\mathbf{x}_q$ as: $\mathbf{y}_{ij}^{d} = (\mathbf{M}_{pca}^{d})^T \mathbf{x}_{ij}$, $\mathbf{y}_{q}^{d} = (\mathbf{M}_{pca}^{d})^T \mathbf{x}_{q}$.

5. Estimate the reference PCA-SPN for camera C_j by: $\mathbf{y}_{C_j}^{d} = \frac{1}{L} \sum_{i=1}^{L} \mathbf{y}_{ij}^{d}$.

6. Calculate the NCC value $\rho(\mathbf{y}_{q}^{d}, \mathbf{y}_{C_j}^{d})$ between query $\mathbf{y}_{q}^{d}$ and each reference $\mathbf{y}_{C_j}^{d}$ using Eq. (2.14).

7. Accept H_0 if $\rho(\mathbf{y}_{q}^{d}, \mathbf{y}_{C_j}^{d}) < \tau_{\mathbf{y}}$, otherwise accept H_1.

With the feature extractor $\mathbf{M}_{lda}$, a $(c-1)$-dimensional vector $\mathbf{z}$ can be obtained

$$\mathbf{z} = \mathbf{M}_{lda}^{T} \mathbf{y}^{d} = \mathbf{M}_{lda}^{T} (\mathbf{M}_{pca}^{d})^T \mathbf{x}, \tag{4.12}$$

where $\mathbf{z}$ is another compact version of the noise residual $\mathbf{x}$.

4.2.4 Source camera identification using the Proposed Method

The camera identification process using the proposed compact features are summarized in the following Algorithm 4.1 and 4.2. We call the feature vector $\mathbf{y}^d$ and $\mathbf{z}$ produced by Algorithm 4.1 and 4.2 as "PCA-SPN" and "LDA-SPN" respectively

Algorithm 4.2. LDA-based Source Camera Identification

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

5. Use the PCA-SPNs $\{\{\mathbf{y}_{ij}^d\}_{i=1}^L\}_{j=1}^c$ as training samples to estimate the transformation matrix $\mathbf{M}_{lda}$ using Eq. (12).

6. Extract LDA-SPNs from all the training samples $\{\{\mathbf{y}_{ij}^d\}_{i=1}^L\}_{j=1}^c$ and the query $\mathbf{y}_q^d$ as: $\mathbf{z}_{ij} = \mathbf{M}_{lda}^T \mathbf{y}_{ij}^d$, $\mathbf{z}_q = \mathbf{M}_{lda}^T \mathbf{y}_q^d$.

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. (2.14).

8. Accept H_0 if $\rho(\mathbf{z}_q, \mathbf{z}_{C_j}) < \tau_{\mathbf{z}}$, otherwise accept H_1.

in the rest of this chapter. As mentioned earlier, the complexity of calculating correlation is proportional to the dimensionality of the features. 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 the original noise residual $\mathbf{x} \in \mathbb{R}^m$, using either $\mathbf{y}^d$ or $\mathbf{z}$ as the replacement of the original $\mathbf{x}$ would lead to an improvement of roughly 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 [58], respectively.

4.3 Experiments

In this section, we carry out experiments on the Dresden image database [76] to validate the feasibility of the proposed methods. First we evaluate and discuss some main parameters, which play key roles in the proposed method. Significant perform gain is achieved by using the proposed training construction process, which can suppress the unwanted noises. After that we plot the histogram of intra-class and inter-

class correlations to demonstrate the effectiveness of PCA/LDA features. Based on several popular SPN algorithms, we also use our method as a post-processing framework, and the experimental results demonstrate substantial performance gain in terms of overall ROC curve. We also consider some real-world scenarios with limited training data, and find that we may take advantage of the contaminated images as training data when there are limited number of flatfield reference images. Finally, the performance in terms of computational efficiency of the proposed methods are also reported.

4.3.1 Experimental Setup

In this work, images taken by 36 cameras from Dresden image database are chosen and used. As listed in Table 4.1, we can see these 36 cameras are from 15 different models and each model may have 1 to 5 devices. A total of 7200 images from these 36 cameras are involved in our experiments. Each camera has 200 images, including 150 images with varying scene details (i.e., contaminated images) and 50 flatfield images. In our experiments, we consider two scenarios with different types of reference images (i.e., flatfield/contaminated reference), as shown in Table 4.2. All the images used in this work are in 512×512 pixels, and unless state otherwise, we use image blocks cropped from the center of the full size images so as to the avoid the vignetting effect [77]. For each image block, we extract 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 as

$$\mathbf{x} = 0.299\mathbf{x}_R + 0.587\mathbf{x}_G + 0.114\mathbf{x}_B, \tag{4.13}$$

Table 4.1: 36 Cameras involved in our experiments

Camera Models	**Num 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 4.2: The setup of two SCI scenarios

Scenario	**Reference Images per Camera**	**Query Images per Camera**
1	50 flatfield images	100 contaminated images
2	50 contaminated images	100 contaminated images

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

In our experiments, the noise residuals extracted by the methods in [6] (Basic), [44] (BM3D), [9] (MLE) and [11] (PCAI8) are served as the original features. SEA [54] is applied to enhance the reference SPNs for the original SPN and the training samples for PCA-SPN and LDA-SPN. The existing SPN compression method (SPN digest [66]) is performed for the algorithm comparison with the proposed methods. NCC defined in Eq. (2.14) is used to measure the similarity in the SCI tasks.

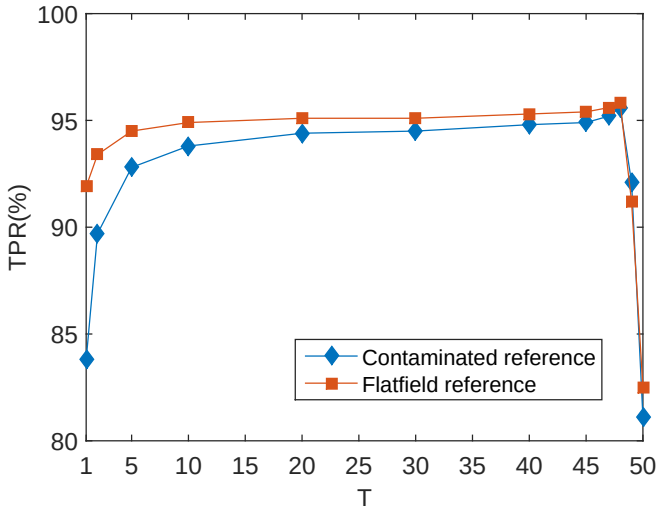


Figure 4.2: The TPR (with the FPR fixed at 10^{-3}) of the PCA-SPN obtained from BM3D w.r.t. different setting of parameter T and different reference types.

4.3.2 Parameter Settings and Discussion

In this work, one of the most important parameter is the number of noise residuals used to estimate a training sample T (also referred to as the random subset size). As discussed in Section 4.2.1, we set T to a relatively large number (i.e., $T \to N$, and $N = 50$) so as to better attenuate the impact of scene details and random noises. Fig. 4.2 depicts the sensitivity of performance (i.e., True Positive Rate TPR at the False Positive Rate FPR 10^{-3}) w.r.t. T in the two SCI scenarios described in Table 4.2. We can see that generally the performance (based on PCA-SPN from BM3D features) is not very sensitive when T is at the range $[20, 48]$, and it improves with an increasing value of T, reaching the peaks when $T = 48$ for both scenarios, i.e., when with flatfield/contaminated reference. It is worth noting the result with $T = 1$ is the case without training set construction, and the corresponding large performance margin (e.g., when compared with $T = 48$) indicates the effectiveness

Table 4.3: The dimensionality d of PCA-SPNs obtained from different SPN methods w.r.t. different setting of T and different reference types.

Method	**Flatfield**		**Contaminated**	
	$T = 20$	$T = 48$	$T = 20$	$T = 48$
Basic	1042	609	1159	867
MLE	1013	605	1138	863
BM3D	1029	598	1148	848
PCAI8	1066	663	1148	860

of our proposed training construction process, especially for contaminated reference. It is also interesting to see the TPR drops dramatically when $T >= 49$, since when $T \to N$ ($N = 50$), all the obtained 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 [84], which is also experimental validated in Section 4.3.5. For the best effect of unwanted noises suppression, we set T to 48 throughout the rest of this chapter.

It is also very interesting to discuss d, the dimensionality of PCA-SPN in different cases, which is determined by the variance PCA aims to preserve (i.e., 98% in this work). Clearly, we prefer d to be as small as possible compatible with accurate identification. From the experiments, we found that d is determined by three factors, which are the percentage of the total variance retained in Eq. (4.8), the dimensionality of original SPN m, and the quality of the training set. As described in Eq. (4.8), we keep the top d eigenvectors corresponding to the 98% of the total variance. Table 4.3 shows the dimensionality d of the PCA-SPNs obtained from different SPN

extraction methods with respect to different setting of T (i.e., $T = 20$ and $T = 48$), for two types of reference images. In both scenarios (flatfield/contaminated training set), 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.(4.4) we can see the quality of the training set tend to be better (i.e., lower $\sigma^2_{\mathbf{X}'}$), thus the energy of the true SPN signal is more clustered in the transformed domain. As a result, the feature extractor require less leading eigenvectors to cover the 98% of the total energy so that the extracted PCA-SPN consequently has a lower dimensionality. Therefore, a training set with better quality would lead to more compact PCA-SPN. Similarly, flatfield reference (with better training sample quality) also tends to have a much compacter representation than the contaminated counterpart, as shown in Table 4.3.

4.3.3 Distributions of Intra-class and Inter-class Correlations

We evaluate the effectiveness of different features in terms of inter/intra-class correlations, which are histograms measuring the similarities of the match/non-match SPN pairs. A clear separation between intra-class and inter-class distributions indicates a feature with the high discriminant ability. Experiments are conducted using three different features (i.e., original SPN, PCA-SPN, and LDA-SPN) in the two SCI scenarios (with flatfield/contaminated reference as listed in Table 4.2). Results are reported in Fig. 4.3, from which we can see the means of intra-class correlations are significantly increased by using PCA-SPN and LDA-SPN, when compared with the results based on original SPN. Specifically, for the two SCI scenarios, the application of PCA increases intra-class correlation mean from 0.046 to 0.564 for

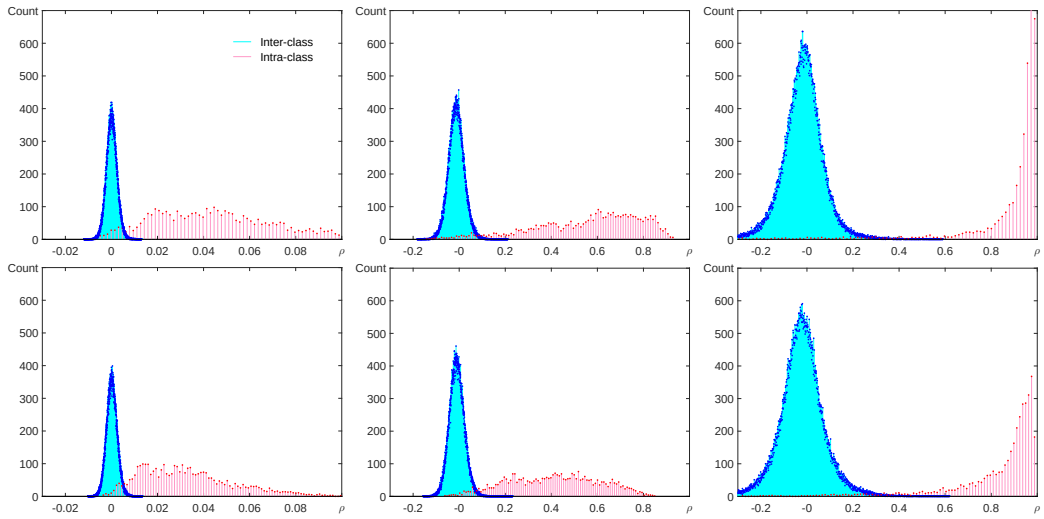


Figure 4.3: Distributions of the inter/intra-class correlations w.r.t. different features (i.e., original SPN, PCA-SPN and LDA-SPN from left to right) and different reference types (1st row: flatfield reference and 2nd row: contaminated reference).

the flatfield reference while from 0.033 to 0.412 when only given the contaminated images as reference. The means of intra-class correlations can be further boosted by LDA-based features owing to its supervised-learning nature, to 0.883 and 0.838, respectively in the two aforementioned scenarios.

The increase of the intra-class correlation means shifts the intra-class distribution rightwards, which contributes positively to a clearer separation between the intra/inter-class similarity distributions. However, the variance of the inter-class correlation is also improved when applying PCA-SPN and LDA-SPN. For example, for the flatfield reference, the inter-class variance for PCA-SPN and LDA-SPN are 7.8×10^{-4} and 6.8×10^{-3}, much larger than that of the original feature, 5.4×10^{-6}. However, they are trivial when compared to the displacements of the means of the intra-class correlation (i.e., $0.564 - 0.046 = 0.518$ and $0.883 - 0.046 = 0.837$) away from the inter-class means, suggesting the benefits of applying PCA-SPN and LDA-SPN on the SCI tasks. This is clearly reconfirmed by in Fig. 4.3, where the

overlapping area between 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, as shown in the first columns of Fig. 4.3, the resulting intra-class correlation distribution of the original feature has small peaks in the overlapping area, which is mainly due to the small negative correlation exhibited among the matching SPN pairs. This small correlation is probably caused by the strong contaminations from scene details in some query images. Nevertheless, as clearly shown in the figures from the last two columns, this small peak is suppressed in the intra-class distribution of PCA-SPN and LDA-SPN. As a result, the overlapping area is decreased substantially. Moreover, since the separation is mainly caused by the right shifting of the intra-class distribution, which has a major influence on the False Rejection Rate (FRR), therefore PCA-SPN and LDA-SPN have particular advantage in the situations where low FRR is preferred.

4.3.4 Comparison of the Overall ROC Curves

We can use the aforemention methods (training construction and feature extraction) as a general post-processing procedure for existing methods. For evaluation, here we report the results in terms of overall ROC curves for four popular SPN algorithms (i.e., Basic [6], BM3D [44], MLE [9] and PCAI8 [11]) combined with and without the proposed post-processing method. Moreover, since our method aims to compress the size of SPN, we also present another SPN compression method, i.e., SPN Digest [66], for the comparison purpose. The SPN digest is primarily formed by retaining the top k largest elements from a m-dimensional SPN ($k < m$). Therefore, the SPN

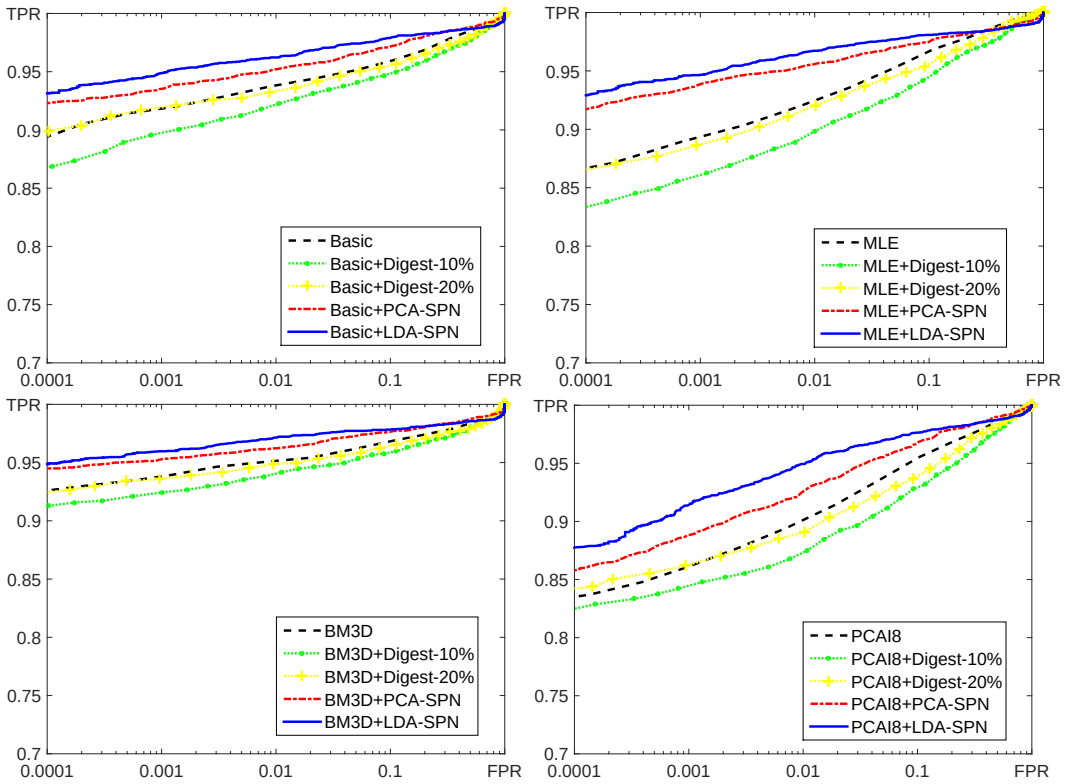


Figure 4.4: Overall ROC curves comparisons among different types of features, i.e., original SPNs, SPN Digests, PCA-SPNs and LDA-SPNs for the contaminated reference.

digest has a dimensionality of k, which is lower than that size of the normal-sized SPN. Moreover, the digest of a reference SPN not only contains k large elements but also the corresponding location information of these elements, which is used to extract a digest from the same locations of each query SPN. In this experiment, we set k/m equal to 10% and 20%.

Fig. 4.4 and Fig. 4.5 show the overall ROC performance for all the methods in the two SCI scenarios described in Table 4.2, i.e., with flatfield/contaminated reference images. In real-world forensic applications, it is often necessary to ensure a sufficiently low FPR, and for highlighting purpose we plot the horizontal axis of the ROC curve in the logarithmic scale. Specifically, the black, green, yellow, red

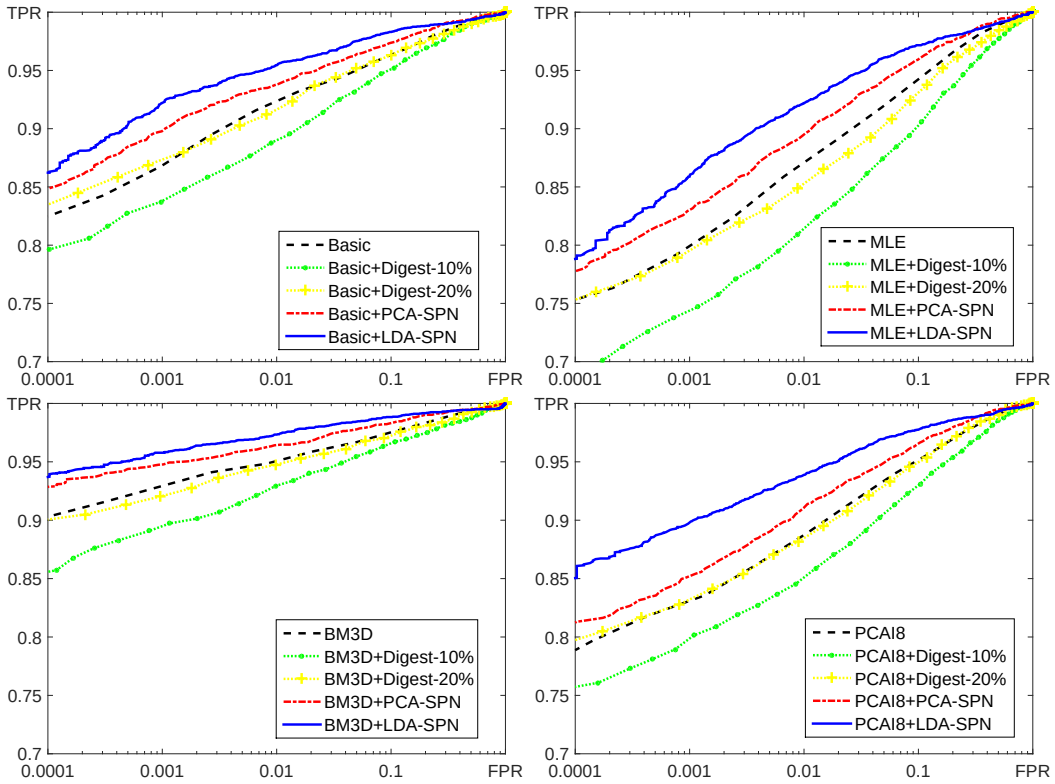


Figure 4.5: Overall ROC curves comparisons among different types of features, i.e., original SPNs, SPN Digests, PCA-SPNs and LDA-SPNs for the flatfield reference.

and blue curves indicate the performance of original SPNs, SPN Digest-10%, SPN Digest-20%, PCA-SPNs and LDA-SPNs, respectively. In both SCI scenarios, we can see SPN Digest has a very close result with the original SPN when 20% of the top largest elements are retained, but its performance degrades rapidly when this amount is reduced to 10%. It indicates a trade-off between compression ratio and identification accuracy for the SPN Digest technique. On the other hand, the LDA-SPNs (blue lines) constantly achieves the best ROC performance regardless of the SPN extraction algorithms and the type of reference images, while the PCA-SPNs (red line) always take the second place. This observation indicates the superiority of our method over SPN Digest on the ROC performance. Moreover, the dimen-

Table 4.4: TPR (%) of different features at the FPR of 1×10^{-3} with different number of flatfield reference images.

		Flatfield			**Contaminated**
	#Ref. per Class:	50	30	10	50
Original	Basic	91.8	90.9	85.7	86.9
	MLE	89.6	88.7	84.3	79.9
	BM3D	94.5	93.7	90.0	92.9
	PCAI8	86.2	85.1	81.6	83.1
Digest	Basic	91.9	91.0	86.1	87.4
	MLE	88.7	88.1	83.1	79.5
	BM3D	94.6	93.7	89.8	92.1
	PCAI8	86.3	84.4	80.8	83.2
PCA-SPN	Basic	93.5	92.8	89.3	89.9
	MLE	93.7	91.7	86.7	83.0
	BM3D	95.8	95.1	92.3	94.8
	PCAI8	88.9	86.8	83.0	85.2
LDA-SPN	Basic	94.9	94.3	90.5	92.3
	MLE	94.6	93.5	88.3	85.9
	BM3D	96.3	95.8	93.0	95.8
	PCAI8	91.2	88.8	84.6	89.8

sionality of SPN Digest-20% for 512×512 blocks are 52429, which is much higher than that of PCA-SPN and LDA-SPN (as shown in Table 4.3). Considering both aspects, we can conclude that both PCA-SPN and LDA-SPN are superior than the SPN Digest in terms of compression ratio and identification accuracy, and the experimental results suggest that it can be used as a general post-processing framework for various SPN methods.

4.3.5 Some Observations in Real-World Scenarios

In controlled environment, normally SCI tasks with the flatfield reference would outperform the contaminated one significantly, as shown in Fig. 4.4 and Fig. 4.5.

However, in real-world scenarios there may exist the case that only a few flatfield reference images (e.g., images from the target user's Facebook account) are available, or not at all. We evaluate some of these cases and report the TPR (at the FPR of 1×10^{-3}) in Table 4.4. We can see the performance degrades rapidly with the decreasing of the flatfield reference. In some extreme cases (when # flatfiled = 10), it may be more beneficial to use contaminated images to train the effective features (e.g., PCA/LDA-SPN) before performing classification. Due to the limitation (only 50 flatfield and 150 contaminated images available per camera) of this dataset, larger-scale experiments are required to evaluate the effectiveness of the use of contaminated images as reference (for training), which will be our future work.

4.3.6 Comparison of Computational Complexity

In this section, we evaluate the efficiency of the SCI system, based on the proposed feature extraction framework. This experiment is performed on a simulated database, which contains 180 cameras derived from the 36 cameras in Table 4.1. To simulate the reference SPNs of 180 cameras, we assume SPNs extracted from two different positions are independent. We first estimate the full-sized reference SPN for each camera in Table 4.1. Then we crop 5 SPN blocks from different locations of each full-sized reference SPN. We deem the 5 cropped SPN blocks as reference for 5 different cameras, so that eventually we obtain 36×5 reference SPNs in total. Table 4.5 shows the running time of matching 500 query noise residuals to the simulated 180 cameras with respect to different types of features. In this case, the size of the original SPN, SPN Digest, PCA-SPN and LDA-SPN are $m = 262144$, $k = 52429$, $d = 2484$ and $c - 1 = 179$, respectively. This experiment is conducted on the same

Table 4.5: 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	2.01	0	365.58	367.59
SPN Digest	1.28	37.51	49.17	87.96
PCA-SPN	52.91	127.41	5.07	185.39
LDA-SPN	2.00	11.42	3.06	16.48

Table 4.6: 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 MB	344.61 MB
SPN Digest	68.91 MB	68.91 MB
PCA-SPN	2062.03 MB	1.23 MB
LDA-SPN	342.61 MB	0.24 MB

PC with an Intel Core i5 3.20GHz processor and 16G RAM. 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 feature extractor into memory for processing. The second one is "Feature Extraction", indicating the time spend on extracting SPN Digest, PCA-SPNs or LDA-SPNs from 500 query noise residuals. The third factor is the time cost of calculating the similarity between the 500 queries and the 180 references, which is referred to as "Matching". The overall computational cost is presented as "Total".

As shown in Table 4.5, PCA-SPN incurs the highest computational cost in I/O operations. It is because the data needs to be loaded into memory not only includes the 180 m-dimensional reference vectors but also an $m \times d$-dimensional

feature extractor. As shown in Table 4.6, PCA-SPN needs a relatively small space to store its 180 reference vectors (1.23 MB) but a huge space for the feature extractor (2062.03 MB). With such a huge amount of data in total, it is not surprising to see that PCA-SPN has the highest cost in I/O operations. LDA-SPN also need to load a feature extractor into memory, 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 342.61 MB. Moreover, since the size of LDA-SPN is only $c-1$, it needs a negligible space for the 180 reference vectors (0.24 MB). Although in this experimental setting, the total storage requirement of LDA-SPN (342.85 MB) is only slightly lower than that of the original SPN (344.61 MB), it is obvious that this margin will grow in a linear manner w.r.t. increasing number of cameras.

SPN Digest has the smallest storage requirement among these 4 types of features. As mentioned earlier, a digest of a normal-sized SPN consists of not only the k top largest elements but also the corresponding position information of these k elements. This location information will be used to extract a digest from each query SPN so that the location information of each reference digest can be seen as a feature extractor. Therefore, when using SPN Digest, the data needs to be loaded includes not only 180 k-dimensional reference digests but also 180 corresponding k-dimensional feature extractors, which take up a space of 137.82 MB in total. This observation indicates the superiority of SPN Digest in terms of storage requirement. As a result, SPN Digest achieves the lowest computational cost in I/O operations.

As mentioned in [64], 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 in the database and the dimensionality of each feature. Since the

number of references in the database is fixed, LDA-SPN, which has the lowest dimensionality, therefore requires the least computational cost in matching process. PCA-SPN takes the second place, followed by SPN Digest and Original SPN. Although LDA-SPN, PCA-SPN and SPN Digest incur extra computational cost in the feature extraction process, but with all aspects taken into account we can see that replacing Original SPN with LDA-SPN, PCA-SPN or SPN Digest can significantly reduce the overall computational cost. Note that the computational cost of reference estimation, PCA/LDA training and estimating a digest from a reference SPN are not counted in this experiment as all of these process can be performed off-line.

4.4 Conclusion

In this chapter, we introduce and evaluate the concept of PCA de-noising in the SCI task. Based on this concept, an effective framework for de-noising and compressing the full-sized SPN is proposed. For better effect, we also propose a training set construction method that minimizes the impact of the interfering artifacts, which play an important role in learning effective feature extractors that is insensitive to various unwanted noises. 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 on the challenging tasks when only contaminated reference is available, which is usually the case in real-world applications.

Chapter 5

Incrementally Updated Feature Extraction for Source Camera Identification

5.1 Motivation

In our previous chapter, a PCA-based feature extraction (PCAFE) method was proposed to boost the performance of the SPN-based source camera identification system. However, there is a limitation of this method, that the trained feature extractor cannot well represent the cameras (classes) that are not involved in training process. More precisely, to train a feature extractor that can well represent every camera in a database, it requires all the cameras to be available before the training process is performed, otherwise the PCAFE based identification system would fail in identifying the cameras that are not involved in training. Fig. 5.1 shows a difficult

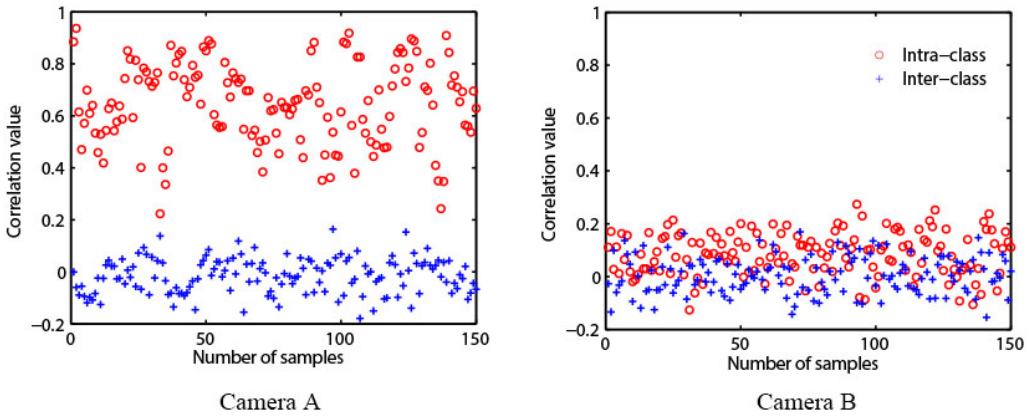


Figure 5.1: Distribution of the intra-class and inter-class correlation values obtained from Camera A (covered in training) and B (not covered by training) on image block with size of 256×256 pixels.

example which can not be correctly identified by the PCAFE-based system. In this example, Camera A is involved in training, while Camera B is not. By comparing the results in the two figures, we can see that the PCAFE-based system works well when the camera is involved in the training, but fails to separate the intra-class and inter-class correlations when the camera is not covered by training. However, in real-world scenarios, we may face the case that images taken by new cameras are continuously added to the database. In this case, the PCAFE-based system requires to re-conduct a training that includes these new received images so as to maintain the identification accuracy. As mentioned earlier, training is actually a process of computing the eigenvectors and eigenvalues of the sample covariance matrix by using a numerical method, such as the power method and the QR method [85]. Thus, the training process can incur a costly computation and exorbitant memory-requirement burden when sample size and sample number are both large. As a result, it is very inefficient to repeat training every time when new sample arrives. To address this problem, an incremental method is usually applied to compute the

principal eigenvectors for data arriving sequentially, where the estimate of principal eigenvectors are updated by each arriving data. Motivated by this, in this chapter, we propose an incremental learning method to extend the PCAFE based system so as to accommodate the new images and update the feature extractor incrementally.

5.2 Problem Statement

The topic of incremental learning has been studied for more than two decades by the machine learning community. In the literature, many incremental learning methods [86–93] have been proposed to incrementally update eigenvectors without estimating the covariance matrix. These methods take the samples sequentially and compute the new set of eigenvectors based on the previous eigenvectors and the new received samples. Among these methods, candid covariance-free incremental principal component analysis (CCIPCA), introduced in [90, 91], is known to be the fastest in convergence rate and lowest in computational complexity especially for high-dimensional data, which maybe an efficient solution in our case.

Unlike the common features with a structural outline (e.g., human face), SPN is a kind of noise-like signal which has more random and complex variations. Therefore, it is impossible to compact its energy into a few number of principal components. In fact, hundreds of eigenvectors are normally required to cover the variations of an even small-sized SPN database. Moreover, the generation of new SPN samples may cause an increase in the variations of the database. As a result, a representative feature extractor would require more eigenvectors to capture the variations after new SPN samples of new class arrive. As shown in Table 5.1, one can see that the number of eigenvectors increases when the number of cameras/classes

Table 5.1: Dimensionality of PCA-SPN with respect to the number of cameras involved in training process on image blocks with size of 512×512 pixels.

	Number of cameras			
	4	6	8	10
Dimensionality	124	186	248	307

in the database are increased.

However, the traditional CCIPCA method [91] performs well on adjusting the direction of the existing eigenvectors, but not on generating new eigenvectors according to the new samples. Geometrically speaking, for a new received data, CCIPCA would first pull the existing eigenvectors toward the direction of new data by a small amount [91]. Then, it would simply absorb the new data as a new eigenvector. By dong so, this new generated eigenvector actually would overfit the current new sample. As a result, it would be difficult for the following new samples to pull it back. In the view of this limitation, we propose a method to improve CCIPCA so as to estimate a more reliable feature extractor according to the incrementally received images.

5.3 Proposed Method

Assume that we have already trained a feature extractor $\mathbf{M} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_d]$ from an initial database, and there are n new samples $\{\mathbf{x}_i\}_{i=1}^{n}$ continuously received. d is the number of eigenvectors contained by the initial feature extractor $\mathbf{M}$. Without loss of generality, we assume that $\mathbf{x}_i$ has a zero mean (the mean is incrementally estimated and subtracted out). Given a new sample $\mathbf{x}_i$, we first match it with the classes in

the initial database. If it is detected as an unknown class, we would perform two steps to update the eigenvectors of the initial feature extractor $\mathbf{M}$.

In the first step, the traditional CCIPCA method [91] is applied to update the existing eigenvectors according to the new samples. For each new sample, each eigenvector of the initial feature extractor $\mathbf{M}$ would be updated once, thus they would be updated n times in total. In the rest of this chapter, $\mathbf{M}_i$ and $\mathbf{v}_1(i)$ indicate the feature extractor and the 1st eigenvector after updated by the first i samples, respectively. Since the eigenvectors should be orthogonal to each other, we generate the new sample in a complementary space to update the eigenvectors so as to keep orthogonality [91]. For example, when the i-th sample $\mathbf{x}_i$ arrives, in order to update the 1st eigenvector $\mathbf{v}_1(i-1)$ of the current feature extractor $\mathbf{M}_{i-1}$, we first subtract from the sample $\mathbf{x}_i$ its projection on $\mathbf{v}_1(i-1)$ as follows

$$\begin{aligned}\mathbf{x}_i(1) &= \mathbf{x}_i(0) - \mathbf{v}_1(i-1)[\mathbf{v}_1(i-1)^T\mathbf{x}_i(0)] \\ &= \mathbf{x}_i(0) - \mathbf{x}_i(0)^T\mathbf{v}_1(i-1)\mathbf{v}_1(i-1),\end{aligned} \tag{5.1}$$

where $\mathbf{x}_i(0) = \mathbf{x}_i$. The obtained residual, $\mathbf{x}_i(1)$, which is in the complementary space of $\mathbf{v}_1(i-1)$, would then be used to update the 1st eigenvector $\mathbf{v}_1(i-1)$ of the current feature extractor $\mathbf{M}_{i-1}$. According to [91], the way to update the 1st eigenvector $\mathbf{v}_1(i-1)$ according to the obtained residual $\mathbf{x}_i(1)$ can be expressed as

$$\mathbf{u}_1(i) = \frac{d_{i-1}+l}{d_{i-1}+1}\mathbf{v}_1(i-1) + \frac{1-l}{d_{i-1}+1}\mathbf{x}_i(1)\mathbf{x}_i(1)^T\mathbf{v}_1(i-1) \tag{5.2}$$

and

$$\mathbf{v}_1(i) = \frac{\mathbf{u}_1(i)}{\|\mathbf{u}_1(i)\|}, \tag{5.3}$$

where $\mathbf{v}_1(i)$ indicates the new estimate of the 1st eigenvector based on the last estimate $\mathbf{v}_1(i-1)$, and the new sample $\mathbf{x}_i$. d_{i-1} is the number of eigenvectors in the current feature extractor $\mathbf{M}_{i-1}$, which is calculated as

$$d_{i-1} = \begin{cases} d_0, & i = 1, \\ d_0 + i - 1, & otherwise, \end{cases} \tag{5.4}$$

where $d_0 = d$. The reason for Eq. (5.4) would be explained in the second step. Noting that in Eq. (5.2)

$$\mathbf{x}_i(1)^T \mathbf{v}_1(i-1) \tag{5.5}$$

is a scalar, therefore

$$\frac{1-l}{d_{i-1}+1} \mathbf{x}_i(1)\mathbf{x}_i(1)^T \mathbf{v}_1(i-1) \tag{5.6}$$

is actually a scaled vector of $\mathbf{x}_i(1)$. According to Eq.(5.2), $\mathbf{u}_1(i)$ is essentially a weighted combination of the last estimate of the 1st eigenvector $\mathbf{v}_1(i-1)$ and the scaled vector of $\mathbf{x}_i(1)$. Therefore, geometrically speaking, $\mathbf{u}_1(i)$ is obtained by pulling the last estimate $\mathbf{v}_1(i-1)$ toward the direction of the new sample $\mathbf{x}_i(1)$ by a small amount. Moreover, in Eq. (5.2), $\frac{d_{i-1}+l}{d_{i-1}+1}$ is the weight for the last estimate $\mathbf{v}_1(i-1)$, and $\frac{1-l}{d_{i-1}+1}$ is the weight for the new sample, where l is a weighting parameter $(0 < l < 1)$. Simply speaking, the last estimate $\mathbf{v}_1(i-1)$ is responsible for the existing samples in database. Therefore, with the presence of l, more weights are assigned to the existing samples so that the effect of the existing samples would not fade out quickly.

After generating $\mathbf{v}_1(i)$, $\mathbf{x}_i(2)$ would be calculated (using Eq. (5.1)) to update the 2nd eigenvector $\mathbf{v}_2(i-1)$ so as to estimate $\mathbf{v}_2(i)$ (using Eq. (5.2)) in the next iter-

ation step. By performing the aforementioned progresses to update each eigenvector of $\mathbf{M}_{i-1}$, we can eventually obtain a feature extractor $\mathbf{M}'_i = [\mathbf{v}_1(i), \mathbf{v}_2(i), ..., \mathbf{v}_{d_{i-1}}(i)]$, which would be further updated in the second step.

In the second step, we propose a new method to generate an eigenvector for extending the dimension of the current feature extractor $\mathbf{M}'_i$ according to the new sample $\mathbf{x}_i$. Different from the traditional CCIPCA method [91], which directly keeps the new sample $\mathbf{x}_i(0)$ as a new eigenvector, our idea is to use the residual of sample $\mathbf{x}_i(0)$, which is left in the complementary space of $\mathbf{M}'_i$, as the new eigenvector. This residual contains the information of the new sample $\mathbf{x}_i(0)$, which the current feature extractor $\mathbf{M}'_i$ cannot well represent. So using it as the new eigenvector, the updated feature extractor $\mathbf{M}_i$ (obtained from Eq. (5.9)) would be able to represent the new sample $\mathbf{x}_i(0)$. Moreover, using this residual rather than the new sample $\mathbf{x}_i(0)$ as the new eigenvector would make the updated feature extractor $\mathbf{M}_i$ less overfit the new sample $\mathbf{x}_i(0)$. It would provide us a more accurate estimation on the feature extractor especially when there are several new received samples taken by the same camera. This residual is calculated by subtracting from the sample $\mathbf{x}_1(0)$ by its projection on $\mathbf{M}'_i$, such as

$$\mathbf{r}_i = \mathbf{x}_1(0) - \mathbf{M}'_i(\mathbf{M}'^T_i \mathbf{x}_i(0)). \tag{5.7}$$

The obtain residual $\mathbf{r}_i$ is normalized as

$$\mathbf{r}'_i = \frac{\mathbf{r}_i}{\|\mathbf{r}_i\|}. \tag{5.8}$$

Then we extend the dimension of the current feature extractor $\mathbf{M}'_i$ by keeping the

Algorithm 5.1

Input: the initial feature extractor $\mathbf{M}_0 = [\mathbf{v}_1(0), \mathbf{v}_2(0), ..., \mathbf{v}_d(0)]$, the new received SPN vectors $\{\mathbf{x}_i \in \mathbb{R}^{N^2 \times 1}\}_{i=1}^{n}$;

Output: the updated feature extractor $\mathbf{M}_n$=$[\mathbf{v}_1(n), \mathbf{v}_2(n), ..., \mathbf{v}_d(n)]$;

Initializing d_0: $d_0 = d$;

for $i = 1$ to n **do**

Step 1: Adjust current eigenvectors:

Initializing $\mathbf{x}_i(0)$: $\mathbf{x}_i(0) = \mathbf{x}_i$;

for $k = 1, 2, ..., d_{i-1}$ **do**

$\mathbf{x}_i(k) = \mathbf{x}_i(k-1) - \mathbf{x}_i(k-1)^T \mathbf{v}_k(i-1)\mathbf{v}_k(i-1)$;

$\mathbf{u}_k(i) = \frac{d_{i-1}+l}{d_{i-1}+1}\mathbf{v}_k(i-1) + \frac{1-l}{d_{i-1}+1}\mathbf{x}_i(k)\mathbf{x}_i(k)^T\mathbf{v}_k(i-1)$;

$\mathbf{v}_k(i) = \frac{\mathbf{u}_k(i)}{\|\mathbf{u}_k(i)\|}$;

end for

Step 2. Generate a new eigenvector:

Initializing $\mathbf{M}_i'$: $\mathbf{M}_i' = [\mathbf{v}_1(i), \mathbf{v}_2(i), ..., \mathbf{v}_{d_{i-1}}(i)]$.

Calculating the residual of $\mathbf{x}_i(0)$ in the complementary space of the current $\mathbf{M}_i'$:

$\mathbf{r}_i = \mathbf{x}_i(0) - \mathbf{M}_i'(\mathbf{M}_i'^T \mathbf{x}_i(0))$;

Normalising $\mathbf{r}_i'$: $\mathbf{r}_i' = \frac{\mathbf{r}_i}{\|\mathbf{r}_i\|}$;

Updating $\mathbf{M}_i'$ by adding $\mathbf{r}_i$ as a new eigenvector: $\mathbf{M}_i = [\mathbf{M}_i' \quad \mathbf{r}_i]$;

Updating d_i: $d_i = d_{i-1} + 1$.

end for

normalized $\mathbf{r}_i'$ as a new basis vector, such as

$$\mathbf{M}_i = [\mathbf{M}_i' \quad \mathbf{r}_i'], \tag{5.9}$$

where $\mathbf{M}_i$ is the new estimate of the feature extractor based on the last estimate $\mathbf{M}_{i-1}$ and the new sample $\mathbf{x}_i$. Note that the obtained feature extractor $\mathbf{M}_i$ contains d_i eigenvectors, where $d_i = d_{i-1} + 1 = d_o + i$.

By repeating the aforementioned two steps whenever a new sample arrives, we are able to obtain an updated feature extractor $\mathbf{M}_n = [\mathbf{v}_1(n), \mathbf{v}_2(n), ..., \mathbf{v}_{d+i}(n)]$.

These two steps are summarised in Algorithm 5.1.

In the **Step 1**, $\mathbf{v}_k(0)$ indicates the k-th order eigenvector of the initial feature extractor $\mathbf{M}_0$, and $\mathbf{v}_k(i)$ is the k-th eigenvector after updated by the first i SPN samples. $\mathbf{x}_i(0)$ is the i-th new sample, and $\mathbf{x}_i(k)$ is the i-th sample after subtracted by the projections on the first $k-1$ eigenvectors, i.e., $\mathbf{v}_1(i), \mathbf{v}_2(i), ..., \mathbf{v}_{k-1}(i)$. Comparing to the original method proposed in [91], the main contributions of our method are summarised as follows:

1. We redesign the weighting scheme in Eq. (5.2) so as to make the traditional CCIPCA method [91] feasible for the SPN-based source camera identification. Note that the setting of l actually determines the retaining rate of the old and new data. In Eq. (5.2), the original method [91] proposed to set l as a negative parameter so that more weight would be assigned to the new data. By doing so, although the updated feature extractor would fit the new data faster, the effect of old data would also fade out more quickly. In our case, we prefer the updated feature extractor to represent the new data without diluting the effect of the existing cameras. Therefore, we set $0 < l < 1$ to prevent the system from assigning too much weight to the new sample and diluting the effect of old cameras. Here, we empirically chose $l = 0.6$ via the extensive experiments, Moreover, the feature extractor would be adjusted once whenever a single sample is received. Therefore, in order to avoid too many adjustments caused by a signal camera, we suggest to use at maximum 5 samples from each new camera for updating.

2. With a new sample received, the original method [91] directly absorbs it as a

new eigenvector. By doing so, the new eigenvector would well represent the new sample but probably not the existing cameras in database. To address this problem, we propose to use the residual $\mathbf{r}$ of new sample $\mathbf{x}$ left by the current feature extractor $\mathbf{M}'$ as a new eigenvector. By doing so, firstly it can make sure that the new obtained eigenvector does not overfit the new sample. Secondly, the residual left by $\mathbf{M}'$ is complemented by $\mathbf{M}$, where $\mathbf{M} = [\mathbf{M}' \quad \mathbf{r}]$, thus the obtained $\mathbf{M}$ can represent both old and new data.

5.4 Experiments

5.4.1 Experimental Setup

In this work, the noise residuals extracted by the methods in [7] (MLE) and [11] (PCAI8) are used as the original features. The experimental works are conducted on the Dresden Image Database [76]. A total of 1600 images from 8 cameras are involved in our experiments, each responsible for 200. These 8 cameras belong to 3 camera models, which are listed in Table 5.2. For each camera, we have 50 low-variation images for training and 150 images with scene details for testing. Hence there are 150×8 intra-class and 1050×8 inter-class samples in total. In our experiments, MLE/PCAI8+8C-PCAFE indicates that SPNs are extracted by using MLE/PCAI8, and the feature extractor is trained by using samples from all the 8 cameras in the training set; 5C-PCAFE means the feature extractor is estimated by only using samples from the 5 *existing*-cameras in the training process; and 5(3)C-CCIPCAFE denotes that samples from the 5 *existing*-cameras are first applied to estimate the initial feature extractor and samples from the 3 *added*-cameras are

Table 5.2: 8 Cameras involved in our experiments.

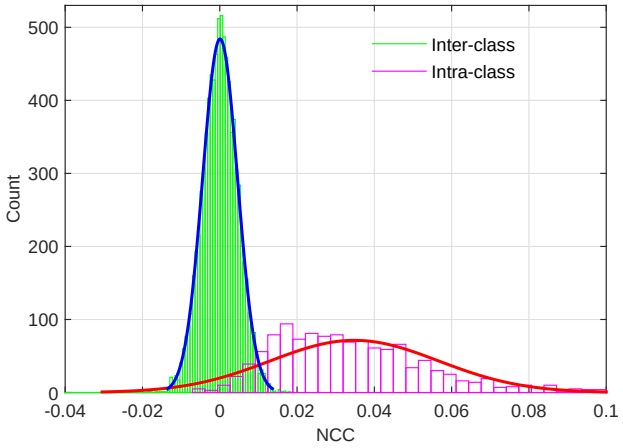
Cameras	**Resolution**	**Satatus**
Canon_Ixus70_A	3072×2304	existing
Canon_Ixus70_B	3072×2304	existing
Nikon_CoolPixS710_A	4352×3264	existing
Samsung_L74wide_A	3072×2304	existing
Samsung_L74wide_B	3072×2304	existing
Canon_Ixus70_C	3072×2304	added
Nikon_CoolPixS710_B	4352×3264	added
Samsung_L74wide_C	3072×2304	added

then sequentially received to update the initial feature extractor via the proposed CCIPCA-based feature extraction (CCIPCAFE) method.

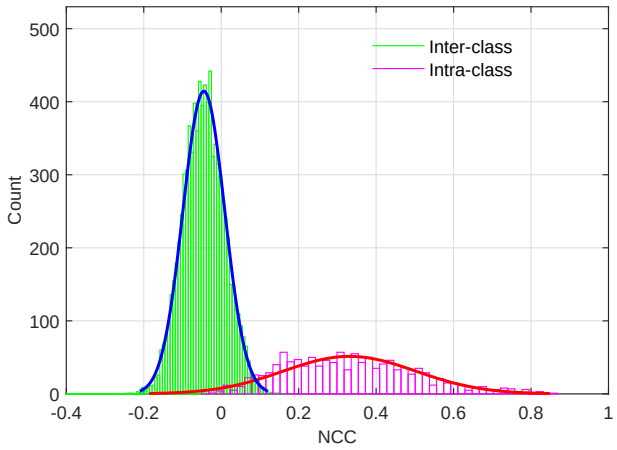
5.4.2 Performance Evaluation

A straightforward way to evaluate the performance of a method is to measure how well it separates the distributions of the intra-class and inter-class samples. Fig. 5.2 shows the histograms of correlation values calculated from features extracted using different methods. Fig. 5.2 (a), (b) and (c) indicate the performance of MLE, MLE+8C-PCAFE and MLE+5(3)C-CCIPCAFE, respectively. Generally speaking, the increase of the intra-class means would shift the intra-class distribution rightwards and contribute positively to a clearer separation between the intra-class and inter-class distributions. By comparing Fig. 5.2 (a) with Fig. 5.2 (b) and Fig. 5.2 (c), we can see that the intra-class mean has been significantly increased after the feature extraction (Notice the difference of the X-axis range present in three figures). Therefore, it suggests that the features extracted by PCAFE and CCIPCAFE are both superior than their original feature.

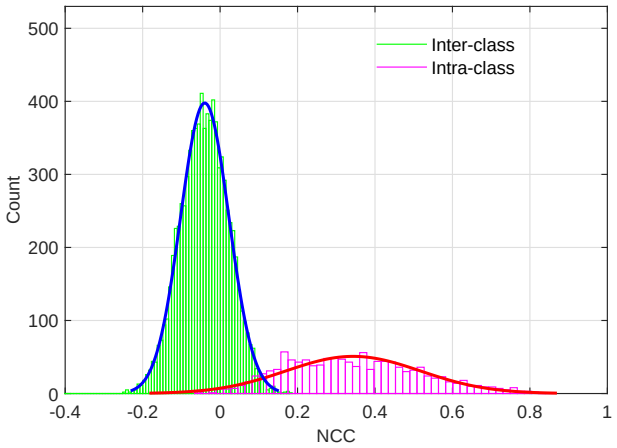
Moreover, the smaller inter-class variance leads to the two distributions more



(a)



(b)



(c)

Figure 5.2: Histograms of the correlation values calculated from features extracted using different methods ((a) MLE, (b) MLE + 8C-PCAFE and (c) MLE + 5(3)C-CCIPCAFE) on the image blocks sized 256×256 pixels.

separable from each other. By comparing Fig. 5.2 (a) and Fig. 5.2 (b), we can see that the inter-class distribution (in green colour) of 5(3)C-CCIPCAFE looks wider than that of 8C-PCAFE in Fig. 5.2 (b), which means 5(3)C-CCIPCAFE has a larger inter-class variance. This observation indicates that PCAFE outperforms CCIPCAFE. While it is not surprising to see CCIPCAFE has a worse performance than PCAFE as estimation error would inevitably occur during updating the feature extractor by using incremental method. In fact, the performance of 8C-PCAFE is the upper bound of 5(3)C-CCIPCAFE.

Next we compare the performance of different methods in terms of the overall receiver operating characteristic (ROC) curve. The corresponding experimental results are shown in Fig. 5.3 and Fig. 5.4, where curves for MLE/PCAI8, MLE/PCAI8+8C-PCAFE, MLE/PCAI8+5(3)C-CCIPCAFE and MLE/PCAI8+5C-PCAFE are highlighted in black, red, blue and green colours, respectively. From these results, we can see that:

1. As expected, both PCAFE (red curves) and CCIPCAFE (blue curves) can boost the conventional SPN extraction methods (black curves) on the overall ROC performance. This is because these two feature extraction methods help to exclude the redundancy and interfering features from the noise residuals that extracted by MLE and PCAI8.

2. The overall ROC performance of 5C-PCAFE is the worst among the four methods. The reason is that the feature extractor is trained by using samples only from the 5 *existing*-cameras, thus this feature extractor is not accurate enough to represent the rest 3 *added*-cameras. As a result, there would be a large number of

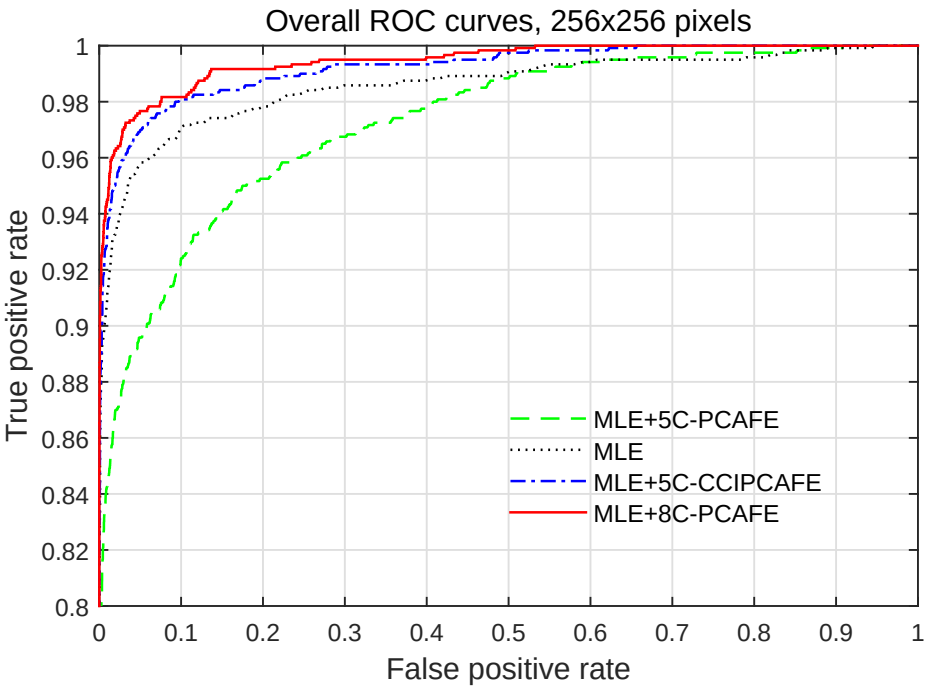


Figure 5.3: The ROC curves of different features based on MLE.

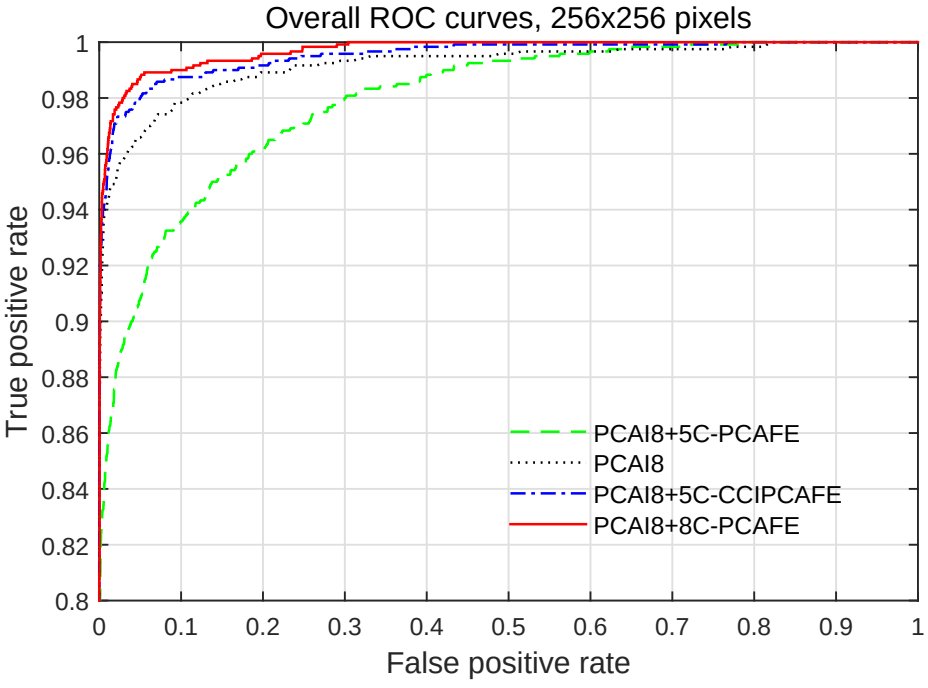


Figure 5.4: The ROC curves of different features based on PCAI8.

Table 5.3: Computational cost (seconds) of different methods on updating a single camera to a database with 10, 20 and 40 cameras, respectively.

	Training time (Seconds)		
	10+1 cameras	20+1 cameras	40+1 cameras
PCA	2.91	9.65	45.84
CCIPCA	0.85	0.86	0.86

false positives from these 3 *added*-cameras. Although repeating a training that includes these 3 *added*-cameras can regain the accuracy, it would incur costly re-computation especially when the number of training sample is huge. Therefore, our CCIPCA-based method is proposed to improve the efficiency. Table 5.3 shows the computational cost of different methods on updating a single camera (with 5 samples). As expected, the proposed CCIPCAFE is much more efficient on updating feature extractor according to new cameras, especially when the size of original database is large.

3. The overall ROC performance of 5(3)C-CCIPCAFE feature is slightly lower than that of 8C-PCAFE. This is a good indication that the proposed incremental updating approach can not only significantly improve the updating efficiency, but also well preserve the identification accuracy.

5.5 Conclusion

In our previous chapter, the PCAFE method was proposed to extract a feature set with much lower dimensionality from the original noise residual. However, this algorithm requires all cameras to be available before feature extractor being estimated. As a result, it would incur costly computation of re-conducting training whenever a

new camera arrives. To solve this problem, an extension based on CCIPCA [91] is proposed to incrementally update feature extractor so as to accommodate the newly received cameras. The experimental results show that both PCAFE and the CCIP-CAFE are able to boost the conventional SPN extraction methods on the overall ROC performance. It not only validates that PCAFE can serve as a post-processing scheme to improve the performance of the conventional SPN extraction methods, but also suggests that when facing a real-time online identification, the CCIPCA-based feature extraction method is an effective extension which can significantly reduce the computational complexity while preserving the identification accuracy.

Chapter 6

Random Subspace Method for Source Camera Identification

6.1 Problem Statement

In Chapter 4, the PCA-based feature extraction (PCAFE) was introduced to extract discriminative features from the noise residuals. As mentioned before, there is a limitation of this method that its performance decreases when the training set is noisy. As we know, SPN can be contaminated by many sources of interferences. When the training samples contain strong interferences (i.e., scene textures), it would be difficult for the training process to capture the variations of true SPN. As a result, the trained eigenvectors are very likely to capture the information of unwanted interferences rather than the true SPN. It is the main reason that the performance of PCAFE decreases. To address this problem, in Chapter 4, we proposed a training set construction method to minimize the impact of the interfering artifacts In this chapter, we propose another solution based on the random subspace method

(RSM) to build a more robust model that generalizes to training data with varying interferences.

6.2 RSM-based Source Camera Identification System

6.2.1 Random Subspace Method

For the learning based methods, there are two common problems: the training data is less representative, and the training set has large dimensionality-to-sample ratio (also known as "curse of dimensionality") [94]. In [95], Ho proposed a random subspace method to address these problems. RSM is defined as an ensemble classifier that consists of several weak classifiers, each operating in a subspace of the original feature space, and outputs the final classification decision by aggregating the results from these individual classifiers. A subspace is constructed by randomly selecting few dimensions from the original feature space, so that each subspace would have much lower dimensionality. Then, a classifier is performed on each subspace for labelling. For each subspace, only the selected dimensions would contribute to the classification result. Since each subspace ignores some dimensions of the original feature space, the corresponding result would be less sensitive to the difference between query samples and training samples in the unselected dimensions [95]. The mislabelling problem may occur in some random subspaces, while the performance of their combination can be much better, especially when the dataset has a large number of features and not too few samples [96]. Therefore, the final classification decision is determined by combining all the decisions associated with the corresponding subspaces. Many ensemble combination rules for combining multiple classifiers

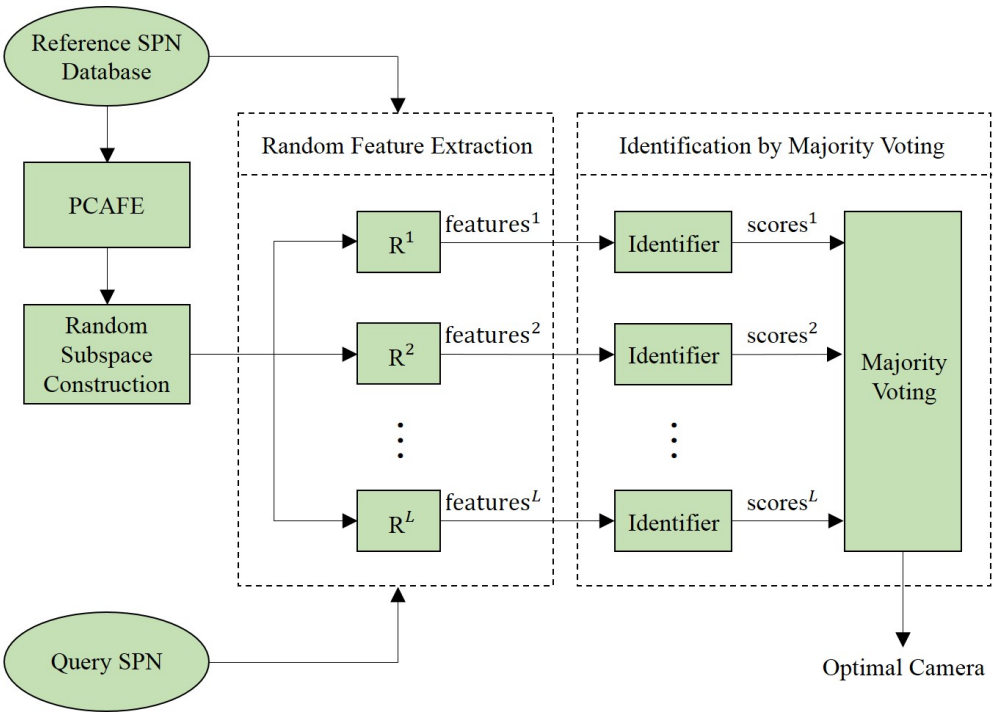


Figure 6.1: The flowchart of the RSM-based SCI system.

have been proposed in [97], such as sum rule [98,99], majority voting [100–102], etc.

The concept of RSM has already been successfully applied in many applications. In [98] and [103], Wang *et al.* employed RSM in the context of face recognition. They proposed to use eigenfaces [104] as candidates to construct random subspaces for random features extraction. In [105], Kuncheva *et al.* employed RSM to classify the functional magnetic resonance imaging (fMRI) data so as to address the overfitting problem caused by the extremely large dimensionality-to-sample ratio. In [100–102], Guan *et al.* proposed an ensemble method based on RSM to reduce the effect of covariate factors for gait recognition. In [106], Marin *et al.* applied RSM to deal with partial occlusions for human detection in still image. In this chapter, we introduce the RSM model in the context of the SPN-based source

camera identification so as to suppress the impact of contaminations in the training samples. The proposed RSM-based SCI system consists of three main stages: random subspace construction, random feature extraction and camera identification by majority voting. A flowchart of the proposed RSM-based SCI system is shown in Fig. 6.1.

6.2.2 Random Subspace Construction

Assume that there are n images $\{\mathbf{I}_i\}_{i=1}^{n}$ taken by c cameras $\{C_j\}_{j=1}^{c}$ in the database and each camera has taken n_j images. The random subspaces can be constructed by the following three steps.

1. We first extract the SPNs from these images by using one of the denoising filters introduced in Section 2.3. The obtained SPNs are utilized as the input SPN samples. Note that SPNs are extracted from $N\times N$-pixels blocks cropped from the centre of the full-sized images. All the obtained SPNs are reshaped into column vectors denoted as $\{\mathbf{x}_i \in \mathbb{R}^{N^2\times 1}\}_{i=1}^{n}$.

2. Due to the high dimensionality of the SPN samples, PCAFE is then applied to reduce the dimensionality. Given the SPN vectors $\{\mathbf{x}_i\}_{i=1}^{n}$, the covariance matrix $\mathbf{S}$ can be estimated as follows

$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T, \tag{6.1}$$

where $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$. The eigenvectors of the matrix $\mathbf{S}$ and the corresponding eigenvalues can be computed via the fast eigen decomposition method mentioned in Section 4.2. Then, the leading $d = \min\{d' \mid \sum_{i=1}^{d'}\lambda_i / \sum_{i=1}^{n}\lambda_i > 98\%\}$ eigenvec-

tors with non-zero eigenvalues are retained as the feature space $\mathbf{M} = [\mathbf{v}_1, ..., \mathbf{v}_d] \in \mathbb{R}^{N^2 \times d}$.

3. Each retained eigenvector in $\mathbf{M}$ is utilized as a candidate eigenvector to build the random subspaces. A random subspace $\mathbf{R}$ can be constructed by randomly selecting m ($m < d$) eigenvectors from these candidates, i.e., $\mathbf{R} = [\mathbf{v}'_1, ..., \mathbf{v}'_d]$, where $\mathbf{v}'$ is the eigenvector randomly selected from $\mathbf{M}$. By repeating L times of randomly selecting subsets from $\mathbf{M}$, L random subspaces $\{\mathbf{R}^l \in \mathbb{R}^{N^2 \times m}\}_{l=1}^{L}$ would be finally generated.

6.2.3 Random Feature Extraction

The obtained random subspaces can be used as random feature extractors. Given a query SPN sample $\mathbf{x}_q$, a low-dimensional SPN feature vector $\mathbf{y}_q^l$ can be extracted by projecting $\mathbf{x}_q$ on the subspace $\mathbf{R}^l$, such as

$$\mathbf{y}_q^l = {\mathbf{R}^l}^T \mathbf{x}_q, \quad l = 1, 2, ..., L. \tag{6.2}$$

These obtained SPN feature vectors $\{\mathbf{y}_q^l\}_{i=1}^{L}$ would be used in the following identification process. Note that the dimensionality of feature is reduced from N^2 to m after the random feature extraction.

To estimate the reference feature for each camera, firstly we perform the random feature extraction (Eq. (6.2)) on all the training samples $\{\mathbf{x}_i\}_{=1}^{n}$. As a result, each training sample $\mathbf{x}_i$ would be represented as a set of features $\{\mathbf{y}_i^l\}_{l=1}^{L}$. Then in the subspace $\mathbf{R}^l$, we can estimate the reference feature $\mathbf{y}_{C_j}^l$ for the camera

C_j by averaging all the features belong to that camera, such as:

$$\mathbf{y}_{C_j}^l = \frac{\sum_{i=1}^{n_j} \mathbf{y}_i^l}{n_j}, \quad j = 1, 2, ..., c, \tag{6.3}$$

where n_j is the number of features belonging to the camera C_j.

6.2.4 Identification by Majority Voting

Based on the obtained query and reference features, the following steps are adopted to decide that the query sample is taken by which camera in the database.

1. In each subspace, an identifier is first applied to match the query sample with all the cameras (classes) in database. Essentially, each matching is a process of verifying whether the query sample is taken by the camera C_j, which can be deemed as a two-channel hypothesis test [55] problem as follows

$$\begin{aligned} &H_0 : \mathbf{y}_q^l \neq \mathbf{y}_{C_j}^l (\text{the query image is not taken by the camera } C_j), \\ &H_1 : \mathbf{y}_q^l = \mathbf{y}_{C_j}^l (\text{the query image is taken by the camera } C_j), \quad j \in [1, c]. \end{aligned} \tag{6.4}$$

   Then a correlation-based detector is established to make the decision between H_0 and H_1 by comparing the correlation $\rho = corr(\mathbf{y}_q^l, \mathbf{y}_{C_j}^l)$ to a pre-determined threshold τ. The detector decides H_1 when $\rho \geq \tau$ and H_0 when $\rho < \tau$. For this type of problem, all the similarity measurements mentioned in Section 2.5 can be used as the detection statistics here. However, considering that several matching processes are required to be performed in each subspace and the number of subspaces can be very large (e.g., $L = 600$), the matching efficiency becomes a crucial factor. As reported in the previous works [9] and [54], the normalized

cross-correlation (NCC) is the most efficient for the SPN matching. Therefore, we adopt NCC here to measure the similarity between the query feature $\mathbf{y}_q^l$ and the reference feature $\mathbf{y}_{C_j}^l$, which is calculated as follows

$$corr\left(\mathbf{y}_q^l, \mathbf{y}_{C_j}^l\right) = \frac{\left(\mathbf{y}_q^l - \overline{\mathbf{y}}_q^l\right)\left(\mathbf{y}_{C_j}^l - \overline{\mathbf{y}}_{C_j}^l\right)}{\left\|\mathbf{y}_q^l - \overline{\mathbf{y}}_q^l\right\|\left\|\mathbf{y}_{C_j}^l - \overline{\mathbf{y}}_{C_j}^l\right\|}, \quad j \in [1, c]. \tag{6.5}$$

where $\|\cdot\|$ is the L_2 norm, $\overline{\mathbf{y}}_q^l$ and $\overline{\mathbf{y}}_{C_j}^l$ are the means of $\mathbf{y}_q^l$ and $\mathbf{y}_{C_j}^l$, respectively. By doing so, the identifier in each subspace would output a decision for each camera on whether the query sample is taken by this camera. By repeating the aforementioned processes in each subspace, L decision would be eventually generated for each camera from all the identifiers.

2. After the first step, we represent the obtain decisions according to a binary function, such as

$$\omega_j^l = \begin{cases} 1, & corr(\mathbf{y}_q^l, \mathbf{y}_{C_j}^l) \geq \tau, \\ 0, & otherwise, \end{cases} \tag{6.6}$$

where ω_j^l is the decision score for the camera C_j, which is generated from the identifier in the subspace $\mathbf{R}^l$. For each camera, we then sum up the corresponding decision scores obtained from all the identifiers, such as:

$$\Omega_j = \sum\nolimits_{l=1}^{L} \omega_j^l, \quad j \in [1, c]. \tag{6.7}$$

The final identification decision can be made through majority voting [107], the

optimal camera C^* is finally chosen as:

$$C^* = \arg\max_j \Omega_j > \frac{L}{2}, \quad j \in [1, c]\,. \tag{6.8}$$

It is worth mentioning that only the camera with a final score $\Omega > L/2$ would be chosen as the optimal camera. If the final scores of all the cameras are lower than $L/2$, the system would determine that the query image was not taken by any cameras in database.

6.3 Experiments

6.3.1 Experimental Setup

In this work, images taken by 10 cameras from Dresden Image Database [76] were chosen and used in our experiments. These 10 cameras belong to 4 camera models, each camera model has 2∼3 different devices. The information of these cameras are listed in Table 6.1. A total of 1500 images from 10 cameras are involved in the experiments, where each camera responsible for 150 images. All these images are natural images containing a wide variety of natural indoor and outdoor scenes. For each camera, 50 images are used for training and the remaining 100 are used as query images for testing. Thus, there are 100×10 intraclass and 900×10 interclass correlation values in total. We extract all the noise residuals from the luminance channel as it contains information of all the three channels. The experiments are performed on the image blocks of three fixed sizes cropped from the centre of the full size images, which are 128×128, 256×256 and 512×512 pixels, respectively.

In this work, the noise residuals extracted by the Basic method [6] and P-

Table 6.1: 10 Cameras involved in our experiments.

Cameras	**Alias**	**Resolution**
Canon_Ixus70_A	C11	3072×2304
Canon_Ixus70_B	C12	3072×2304
Canon_Ixus70_C	C13	3072×2304
Nikon_CoolPixS710_A	C21	4352×3264
Nikon_CoolPixS710_B	C22	4352×3264
Samsung_L74wide_A	C31	3072×2304
Samsung_L74wide_B	C32	3072×2304
Samsung_L74wide_C	C33	3072×2304
Olympus_mju_1050SW_A	C41	3648×2736
Olympus_mju_1050SW_B	C42	3648×2736

CAI8 [11] are used as the original features. NCC, as defined in Eq. (2.15), is used to measure the similarity between the reference feature and the query feature. In order to evaluate the feasibility of the proposed method, these original features are given as inputs for the PCA-based feature extraction method [31] and the proposed method for the performance comparison. Hereafter, Basic/PCAI8+PCA and Basic/PCAI8+RSM indicate that the noise residuals are firstly extracted by Basic/PCAI8 method and the PCA-based feature extraction or the proposed RSM is performed afterwards.

6.3.2 Parameter Settings

There are only two parameters in the proposed method, namely, the dimension of random subspace m and the number of random subspace L. Fig. 6.2 and Fig. 6.3 show how the true positive (false positive) rate of the method Basic+RSM varies for different values of m and L, respectively. As shown in Fig. 6.2, the performance of the proposed method improves by increasing the number of random subspaces L. Since the performance tends to be stable when $L > 400$ and there is a trade-

off between the performance and computational complexity, we set $L = 400$ in the following experiments.

Fig. 6.3 indicates the sensitivity of the proposed method to the parameter m, where m is the dimension of each random subspace and d is the size of the entire feature space T. Note that $m \in [1, d]$ and the performance of the proposed method is as same as that of the PCA-based extraction method [31] when $m = d$. Therefore, from Fig. 6.3 we can see that as long as $m/d < 1$, the proposed method can achieve a higher true positive rate than the PCA-based extraction method. In addition, from both Fig. 6.2 and Fig. 6.3 we can see that the performance of the proposed method is not sensitive to L and M. In the rest of this chapter, we empirically set $L = 400$ and $m/d = 0.45$, because these values yield the best result.

6.3.3 Performance Evaluation

In this work, the overall receiver operating characteristic curve [13] is applied to compare the performance of different methods, as shown in Fig. 6.4, Fig. 6.5 and Fig. 6.6. To get convincing results, all the 100×10 intraclass and $100 \times 10 \times 9$ interclass samples from 10 cameras are used together to draw the overall ROC curve. It is worth mentioning that the overall ROC curve for the proposed method is obtained in a slightly different manner with the one mentioned in Section 2.5. For a given detection threshold, we count the number of true positive decisions and the number of false positive decisions for each camera in each subspace respectively, and sum them up to obtain the total number of true positive decisions and false positive decisions so as to calculate the true positive rate $\mathcal{P}_{tp}$ and false positive rate $\mathcal{P}_{fp}$. Specifically, as the numbers of images captured by each camera are exactly the

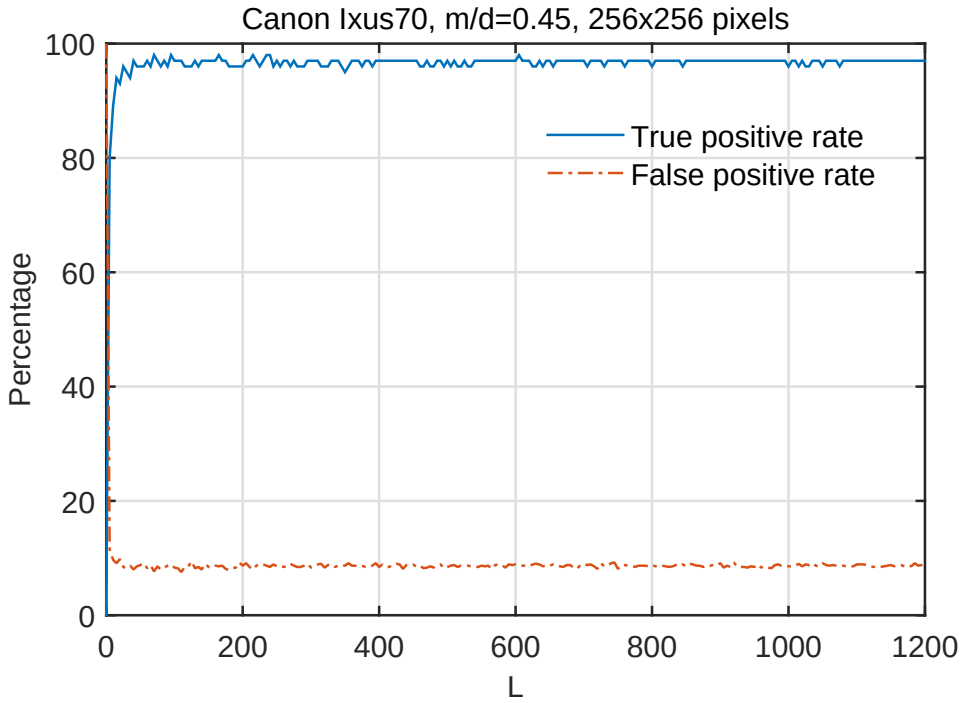


Figure 6.2: Performance with respect to the number of random subspaces L on image blocks with size of 256×256 pixels, threshold $\tau = 0.08$.

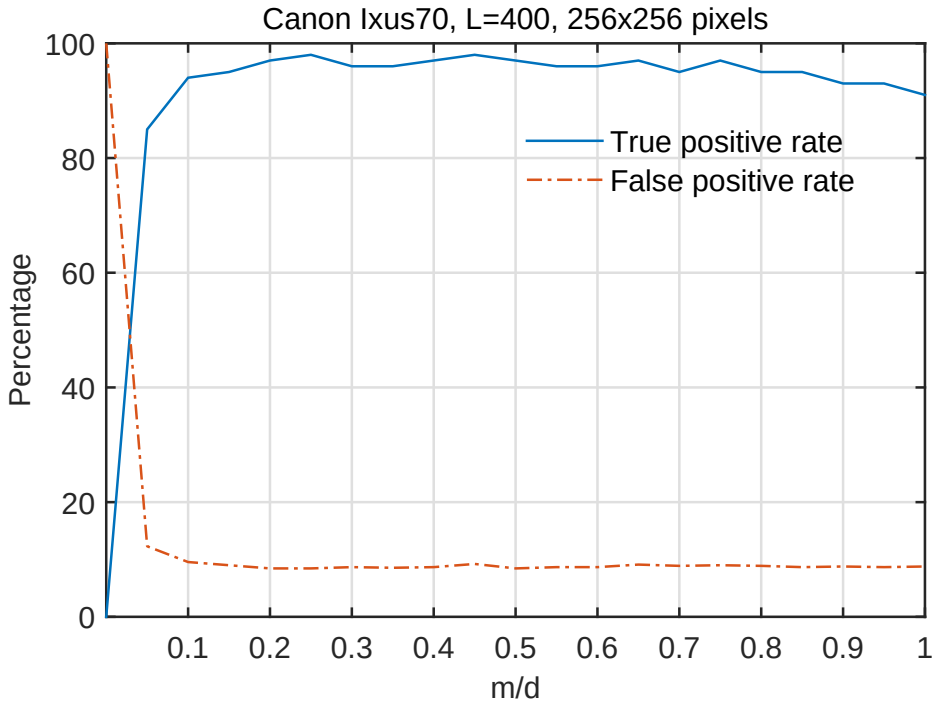


Figure 6.3: Performance with respect to the dimension of the random subspace m on image blocks with size of 256×256 pixels, threshold $\tau = 0.08$.

same, we can simply calculate the $\mathcal{P}_{tp}$ and $\mathcal{P}_{fp}$ for a threshold as follows

$$\mathcal{P}_{tp} = \frac{\sum_{j=1}^{c}\sum_{l=1}^{L}\mathcal{T}_j^l}{T}, \mathcal{P}_{fp} = \frac{\sum_{j=1}^{c}\sum_{l=1}^{L}\mathcal{F}_j^L}{(c-1)T}, i = 1, 2, ..., c, \tag{6.9}$$

where c is the number of cameras, L in the number of subspaces and T is the number of query images from all cameras. $\mathcal{T}_j^l$ and $\mathcal{F}_j^l$ are the true positive decisions and false positive decisions made in the subspace $\mathbf{R}^l$ with respect to camera C_j, respectively. By varying the detection threshold from the minimum to the maximum value, we can finally obtain the overall ROC curve for the RSM-based method.

In Fig. 6.4, Fig. 6.5 and Fig. 6.6, the black, blue, and red lines indicate the overall ROC performance of Basic/PCAI8, Basic/PCAI8+PCA, and Basic/PCAI8+RSM, respectively. In order to show the detail of the ROC curves with a low FPR, the horizontal axis of all the overall ROC curves are plotted in the logarithmic scale. As analysed above, the performance of PCAFE would decrease when the training set is noisy. But from Fig. 6.4, Fig. 6.5 and Fig. 6.6, we can see that as the post-processing method, PCAFE still can boost the performance of the conventional SPN extraction methods even when the training set are full-filled with scene textures. And this performance gain is more significant when PCAFE is performed on the image block with larger sizes (i.e., 512×512 pixels). On the other had, the proposed RSM method (red lines) constantly achieves the best performance regardless of the size of the image blocks and the SPN extraction algorithms. This observation indicates the superiority of the RSM method over PCAFE on the ROC performance. Based on these results, we can conclude that the PCAFE method can improve the conventional SPN extraction methods even when the training set

Table 6.2: Computational cost (Seconds) of different methods on image blocks with size of 512×512 pixels.

Features	**Feature Extraction**	**Matching**	**Total**
Basic	0	4.29	4.29
Basic+PCAFE	2.82	0.03	2.85
Basic+RSM	3.96	19.43	23.39

is noisy, and it can be further improved by the proposed RSM method.

Then we evaluate the computational complexity of the SCI identification system based on the proposed methods. Table 6.2 shows the time cost of different methods to match 100 query noise residuals to the aforementioned 10 cameras on 512×512 image blocks. This experiment is conducted on the same PC with an Intel Core i5 3.20GHz processor and 16G RAM. To quantify the efficiency of an identification system, two factors are considered in this experiment. The first one is "Feature extraction" indicating the time cost for PCAFE and RSM to extract features from 100 query noise residuals and 10 reference SPNs. The second factor is "Matching" which relates to the time spent on calculating 100×10 normalized correlations. The overall computational cost is represented as "Total". The time cost of training and reference estimation are not counted in this experiment as both of them can be performed off-line.

Both PCAFE and RSM require to extract features from the noise residuals, so it is not surprising to see that they require more time in the feature extraction process. The complexity of computing correlation is proportional to the size of the feature vectors. The dimensionality of features extracted by PCAFE is lower than that of the original SPN, thus PCAFE can dramatically reduce the time spent on

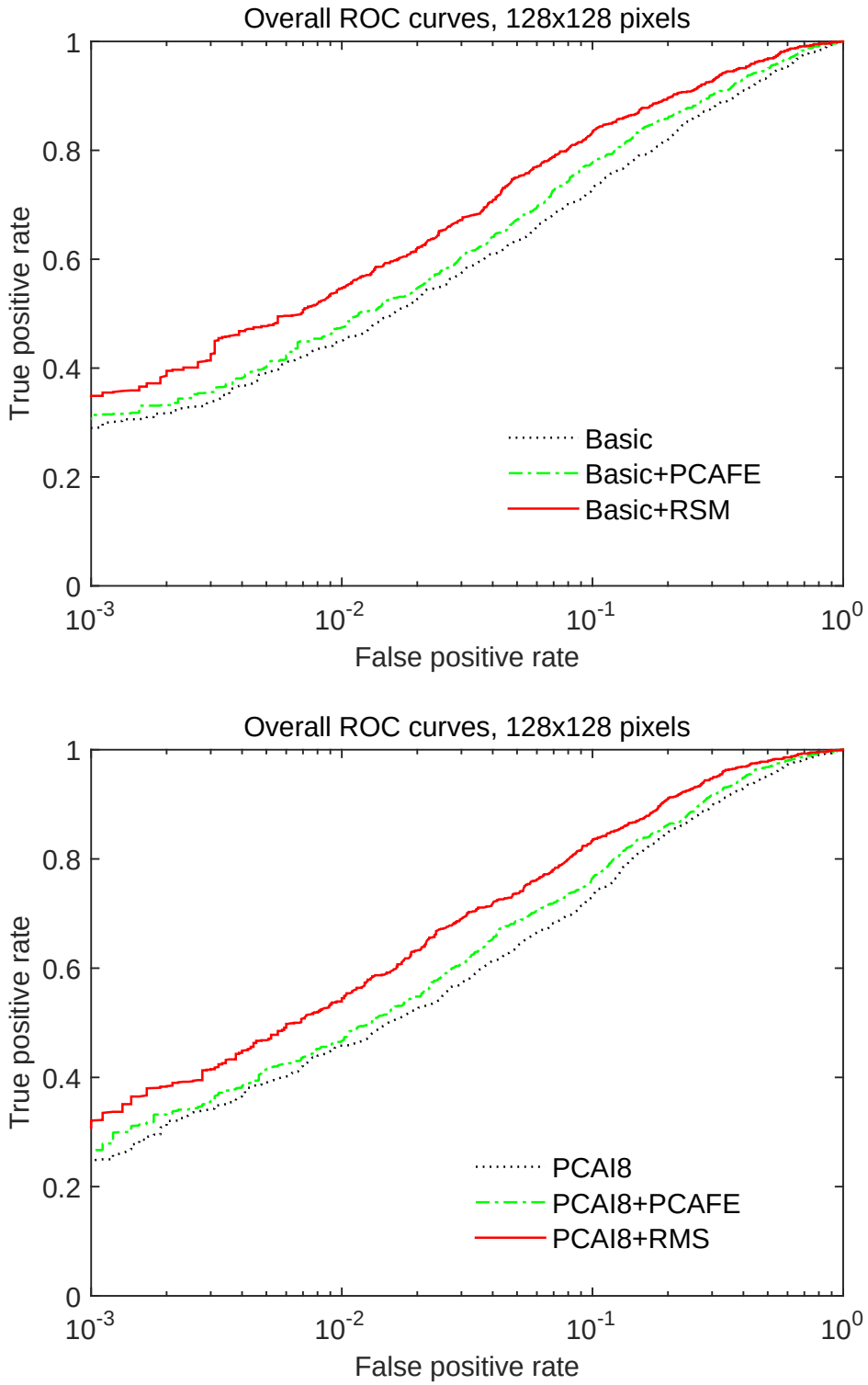


Figure 6.4: The ROC curves of different methods on image blocks with size of 128×128 pixels.

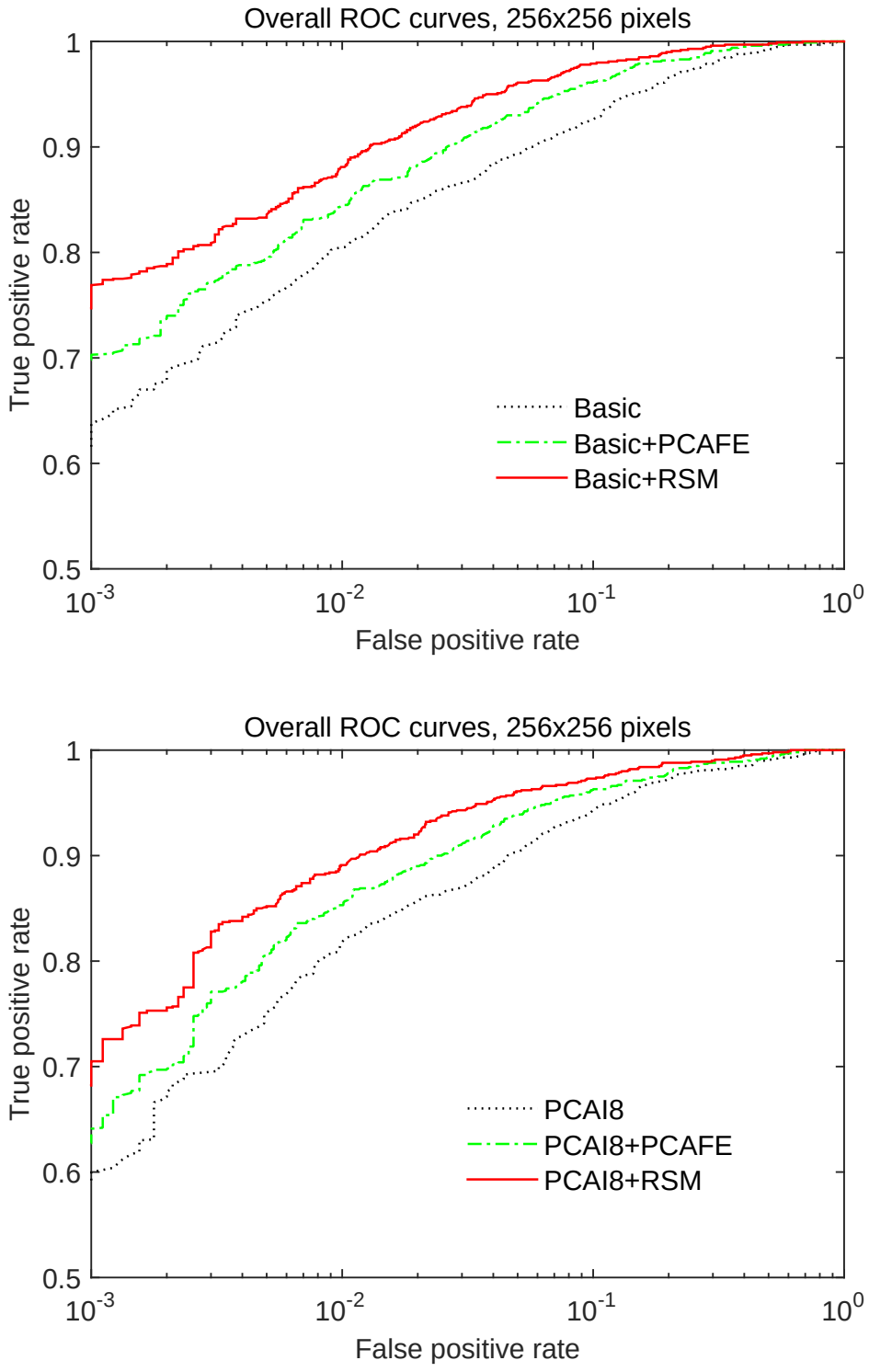


Figure 6.5: The ROC curves of different methods on image blocks with size of 256×256 pixels.

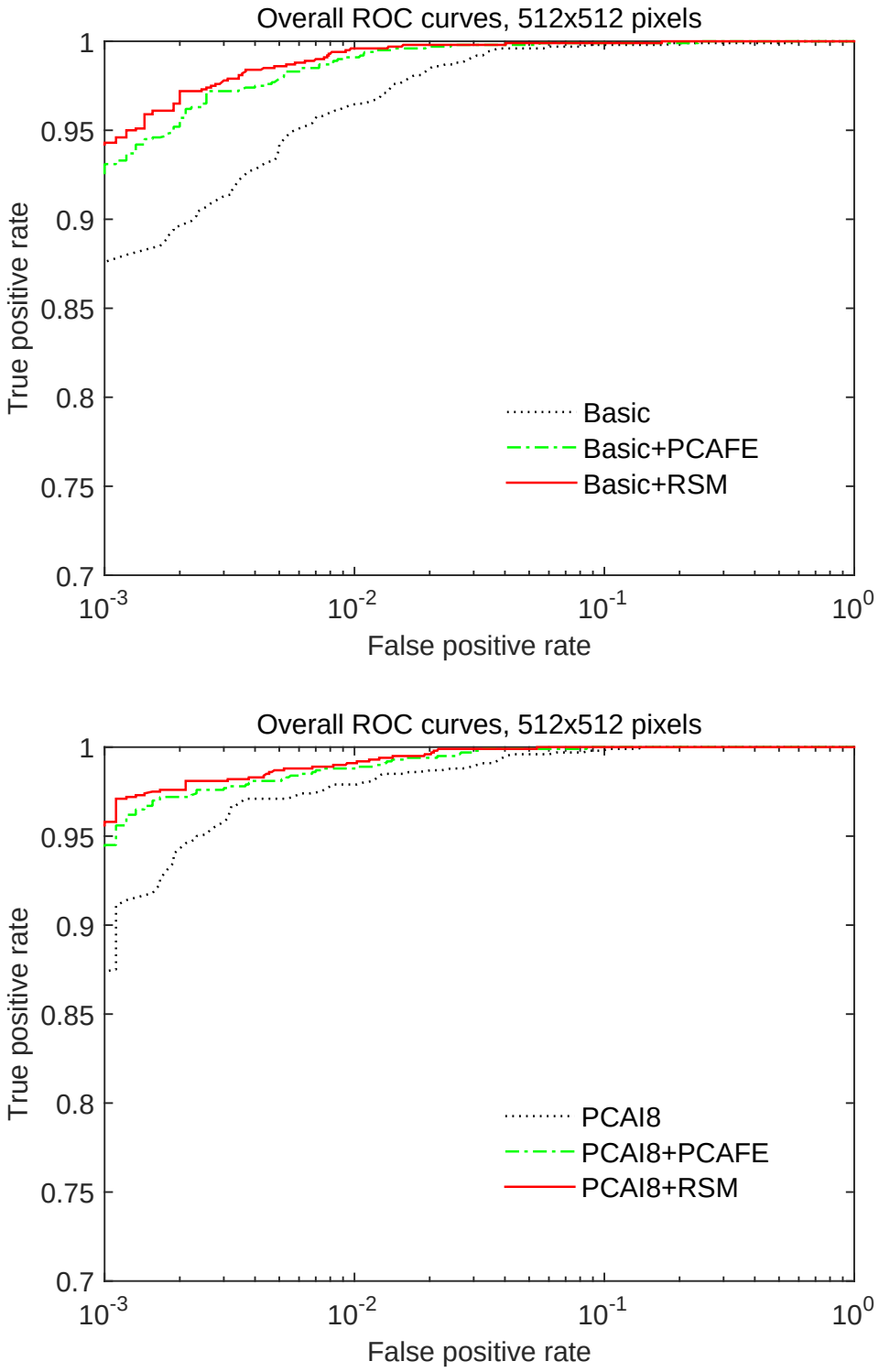


Figure 6.6: The ROC curves of different methods on image blocks with size of 512×512 pixels.

the matching process. The dimensionality of features extracted by RSM m is much lower (i.e., $m/d = 0.45$), but since the RSM method requires to preform matching in each of the 400 subspaces, it is reasonable to see that SEA requires more running time in matching stage. From the overall computational cost, one can deduce that RSM requires more computational cost than other two methods. Although the RSM-based method can bring performance gain to identification accuracy, from this experiment we can see that it also incurs extra cost in computation.

6.4 Conclusion

In Chapter 4, a PCA-based feature extraction method was proposed to extract a feature set with much lower dimensionality from the original noise residual. However, the performance of this algorithm degrades when the training set is noisy. It is because the eigenvectors that generated from the training process can be corrupted by the unwanted interferences. Some leading eigenvectors are very likely to represent the interfering artifacts rather than the real SPN signal. Moreover, it is difficult to locate and remove these corrupted eigenvectors from the feature space. To address these problems, an ensemble solution based on RSM is presented to eliminate the impact of various contaminations. The experimental results show that the proposed RSM-based method achieves a superior overall ROC performance than several SPN extraction methods and the PCA-based feature extraction method. However, the proposed RMS-based method would inevitably bring extra efficiency cost to an SCI systems. Therefore, it suggest that the proposed method is more suitable for the cases that a lower false identification rate is preferred.

Chapter 7

Conclusions and Further Works

7.1 Thesis Summary

In this thesis, we first investigated different methods for the task of source camera verification and identification. In particular, we focused on the sensor pattern noise based matching techniques. SPN is a unique pattern deposited in every image and video taken by a sensor. Due to this uniqueness, SPN can be used as the camera fingerprint. In the past decade, although there has been significant process in the tasks of SPN-based source camera verification and identification, the performance of existing methods is still unsatisfactory in some cases. In this thesis, we considered and addressed two commonly seen but less studied challenges.

The first challenge is the source camera verification with reference images corrupted by scene details. The most significant limitation of using SPN for source camera verification is that SPN can be seriously contaminated by scene details, e.g., textures and edges. Most existing methods assume that the contamination from scene details only occurs in query images but not in reference images, which

is because that forensic investigators normally have access to the camera to be identified. Thus they can use this camera to take some low-variation images, e.g., blue sky images, for the reference SPN estimation. However, this assumption may not hold in the real-world applications, as contamination from scene details may exist in both query and reference images. These two typical contamination cases should be both considered when designing a real-world source camera verification system. Therefore, to address this less studied issue, we proposed a context adaptive reference SPN estimator.

The second challenge we considered is high computational complexity of using SPN in source camera identification, especially when a sizeable database is involved. The SPN-based method has a limitation in source camera identification. This limitation occurs due to the high dimensionality of SPN fingerprints. Many efforts have been made to improve the efficiency of source camera identification in recent years. While these methods can somehow improve the identification efficiency, they also undesirably decrease the identification accuracy at the same time. In order to improve efficiency without degrading accuracy, we propose an effective feature extraction algorithm based on the concept of PCA denoising to extract a small set of principal components from the noisy SPN. We also proposed two extensions, i.e., incrementally updating method and random subspace method, to further improve the performance of this framework. Detailed description of the works involved in this thesis are given as follows:

In Chapter 2, we provided a detailed literature review on the state-of-the-art methods that have been proposed in the SPN-based source camera verification and identification. By analysing the existing methods, we pointed out two challenging

problems which are commonly seen but not well solved.

In Chapter 3, we first proposed a measurement to evaluate the SPN quality of each image block for the reference SPN estimation. This measurement is determined according to both local entropy and local brightness of an image. Based on this measurement, a reference SPN estimator was then proposed to address the aforementioned problem that reference images are contaminated by scene details. Comparing to the traditional method, we proposed to assign higher weights to the SPN blocks with better quality, rather than to treat every SPN block for the reference SPN estimation equally. In the experimental part, we considered the situation that the number of reference images from each camera is inadequate, which is a case that most current works do not take into account. Experimental results showed that the proposed method achieved higher performance in terms of the ROC curves than the methods based on the averaged reference SPN, especially when the number of reference images is inadequate.

In Chapter 4, we introduced the concept of PCA denoising in the task of source camera identification. An effective feature extraction algorithm based on this concept was proposed to compress SPN so as to address the prohibitively computational complexity caused by the high dimensionality of SPN. For better effect, we also propose a training set construction method that minimizes the impact of the interfering artifacts, which play an important role in learning effective feature extractors that is insensitive to various unwanted noises. Moreover, considering that investigators normally have the access to the label information of the database, an enhancement approach based on LDA was adopted to take the advantage of the availability of label information to better separate different classes and further

reduce the dimensionality. Experimental results showed that this method not only significantly improved the efficiency of source camera identification systems but also boosted the performance of several SPN extraction methods on identification accuracy. It suggested that the proposed PCA-based feature extraction (PCAFE) method can be used as a general post-processing scheme for the conventional SPN extraction methods to further improve the SCI system.

In Chapter 5, we proposed an extension of the PCAFE method for incrementally updating feature extractor so as to accommodate the newly received images. The PCAFE method has a major limitation that it cannot well represent the classes (cameras) which are not covered by the training set. In other words, a representative feature extractor can be estimated only if all the cameras have samples involved in the training process. When images taken by new cameras are continuously added to the reference image set, PCAFE requires to re-conduct training so as to ensure the identification accuracy. However, this process is computationally exorbitant. In order to avoid costly computation of re-conducting training whenever a new image arrives, we proposed an extension based on the candid covariance-free incremental PCA (CCIPCA) to incrementally update feature extractor according to the new received images. Compared with the PCAFE method, the proposed CCIPCA-based feature extraction method can significantly reduce the computational complexity for updating new samples with only a slight decrease in the identification accuracy.

In Chapter 6, to further suppress the impact of various interferences in training samples, we proposed a solution based on the random subspace method. In this algorithm, we first applied PCAFE to estimate a feature space that can better represent SPN. Then we constructed a large number of random subspaces by

randomly splitting the obtained feature space into subsets. In each subspace, we performed an identification process and record the identification result. Majority voting was finally applied to make the final identification decision according to the results obtained from all subspaces. Experimental results showed that the proposed RSM based method outperformed the PCAFE method when dealing with a noisy training set.

7.2 Future Works

In this thesis, we have done some works for addressing the challenges in the SPN-based source camera verification and identification system, while more works are yet to be done in order to further improve them. Here we list some possible new lines of investigation for future research.

1. In the proposed PCAFE method, the full sized SPN is used to train the feature extractor. As a result, directly decomposing the covariance matrix is too computationally complex to be performed, thus we have to apply a fast eigen decomposition method. When the number of training samples is large, this fast method can also be inefficient. Therefore, we plan to divide every full sized SPN training sample into many non-overlapping SPN blocks. Then we can gather the SPN blocks from the same location but different training samples and use them to train a local feature extractor. By doing so, we actually divide the original covariance matrix into many small matrices and decompose them respectively, which is a possible way to reduce the computational cost and memory burden. For each location, we can perform a verification by using the query and reference

features extracted by the corresponding local feature extractor. Then, majority voting can be applied to make the final decision according to the results obtained from all locations. Moreover, we can also apply the measurement that we proposed in Chapter 3 to evaluate the SPN quality of each block from the query SPN. Based on the obtained quality score of each query SPN block, we can assign a corresponding confidence to the decision obtained from that SPN block so as to achieve a more accurate identification.

2. In the proposed RSM based identification system, the extracted random features are directly used for identification. However, these extracted features may be redundant and less discriminant. It is because the random feature extractors are trained in an unsupervised manner, without using the label information. As a result, these extracted features may lead to low performance in terms of both computational cost and identification accuracy. These leave room for improvement. Our further plan is to take advantage of the label information of a database and employ supervised learning method, such as uncorrelated LDA (ULDA) [108,109] and IDR/QR [79], for each subspace so as to extract more discriminant features.

3. One of the disadvantages of SPN is that its detection is very sensitive to synchronization [22]. For example, if the image under investigation has been slightly cropped or scaled, the direct verification would not succeed. To address this problem, a brute force search is required to estimate the re-sampling parameters so as to invert the geometrical transformation and then detect the SPN fingerprint. However, the complexity of brute force search can be exorbitant. Although the technique described in [110] may help us to alleviate the computational com-

plexity of brute force searches, the searches would inevitably increase the FAR. In the view of this limitation, we will explore new ways to address this problem. For example, the problem of an SPN saved in different resolutions can be viewed as the problem of an object with multiple views. In many computer vision applications, the same object can be observed at different viewpoints, such as human face and gait [111, 112]. However, the observations from different viewpoints can not be compared directly since the samples from different views may lie in quite different spaces. To address this problem, a Multi-view Discriminant Analysis (MvDA) method [113, 114] was recently proposed to obtain one common space where observations from two different views can be compared. Thus, it can be a possible solution to tackle the problem that images are saved in different resolutions or even compressed with different JPEG quality factors, which is worthy of further research.

Bibliography

[1] H. Pearson, "Image manipulation: CSI: cell biology," *Nature*, vol. 434, no. 7036, pp. 952–953, 2005.

[2] M. J. Sorrell, "Digital camera source identification through JPEG quantisation," in *Multimedia Forensics and Security*, C.-T. Li, Ed. Hershey, PA: IGI Global, 2009.

[3] Camera & Imaging Products Association, "Exchangeable image file format for digital still cameras: Exif version 2.3," Tech. Rep. CIPA DC-008-2010 & JEITA CP-3451B Standard, 2010.

[4] P. Blythe and J. Fridrich, "Secure digital camera," in *Digital Forensic Research Workshop*, 2004, pp. 11–13.

[5] J. Lukas, J. Fridrich, and M. Goljan, "Digital "bullet scratches" for images," in *IEEE International Conference on Image Processing*, 2005, vol. 3, pp. 65–68.

[6] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Transactionss on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, 2006.

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

[8] G. Chierchia, S. Parrilli, G. Poggi, C. Sansone, and L. Verdoliva, "On the influence of denoising in PRNU based forgery detection," in *ACM Workshop on Multimedia in Forensics, Security and Intelligence*, 2010, vol. 7880, pp. 117–122.

[9] 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.

[10] C.-T. Li and Y. Li, "Color-decoupled photo response non-uniformity for digital image forensics," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 2, pp. 260–271, 2012.

[11] X. Kang, J. Chen, K. Lin, and A. Peng, "A context-adaptive spn predictor for trustworthy source camera identification," *EURASIP Journal Image and Video Processing*, vol. 2014, no. 1, pp. 1–11, 2014.

[12] G. Wu, X. Kang, and K. R. Liu, "A context adaptive predictor of sensor pattern noise for camera source identification," in *IEEE International Conference on Image Processing*, 2012, pp. 237–240.

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

[14] T. Filler, J. Fridrich, and M. Goljan, "Using sensor pattern noise for camera model identification," in *IEEE International Conference on Image Processing*, 2008, pp. 1296–1299.

[15] S. Choi, E. Y. Lam, and K. K. Y. Wong, "Source camera identification using footprints from lens aberration," in *SPIE Digital Photography II*, 2006, vol. 6069.

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

[17] S. Bayram, H. Sencar, N. Memon, and I. Avcibas, "Source camera identification based on CFA interpolation," in *IEEE International Conference on Image Processing*, 2005, vol. 3, pp. III–69.

[18] A. Swaminathan, M. Wu, and K. R. Liu, "Nonintrusive component forensics of visual sensors using output images," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 1, pp. 91–106, 2007.

[19] K. Choi, E. Y. Lam, and K. K. Y. Wong, "Source camera identification by JPEG compression statistics for image forensics," in *IEEE Region 10 Conference*, 2006, pp. 1–4.

[20] M. Kharrazi, H. Sencar, and N. Memon, "Blind source camera identification," in *IEEE International Conference on Image Processing*, 2005, pp. 69–72.

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

[22] M. Goljan and J. Fridrich, "Camera identification from scaled and cropped images," in *SPIE, Electronic Imaging, Forensics, Security, Steganography, and Watermarking of Multimedia Contents X*, 2008, pp. OE–1–0E–13.

[23] G. C. Holst, *CCD Arrays, Cameras, and Displays*, vol. 38, Bellingham, WA: SPIE Optical Engineering Press.

[24] J. R. Janesick, *Scientific Charge-coupled Devices*, vol. 117, Bellingham, WA:SPIE press.

[25] J. R. Janesick, "Dueling detectors," *Spie's OE Magazine*, vol. 3033, 2002.

[26] D. Zhang, R. Lukac, X. Wu, and D. Zhang, "PCA-based spatially adaptive denoising of CFA images for single-sensor digital cameras," *IEEE Transactions on Image Processing*, vol. 18, no. 4, pp. 797–812, 2009.

[27] C-A. Deledalle, J. Salmon, and A. S. Dalalyan, "Image denoising with patch based PCA: Local versus global," in *BMVC*, 2011, pp. 1–10.

[28] L. Zhang, W. Dong, D. Zhang, and G. Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," *Pattern Recognition*, vol. 43, no. 4, pp. 1531–1549, 2010.

[29] R. Li, C.-T. Li, and Y. Guan, "A reference estimator based on composite sensor pattern noise for source device identification," in *SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics*, 2014, vol. 9028.

[30] R. Li, Y. Guan, and C.-T. Li, "PCA-based denoising of sensor pattern noise for source camera identification," in *IEEE China Summit & International Conference on Signal and Information Processing*, 2014, pp. 436–440.

[31] R. Li, C.-T. Li, and Y. Guan, "A compact representation of sensor fingerprint for camera identification and fingerprint matching," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Brisbane, Australia, Apr. 19-24, 2015, pp. 1777–1781.

[32] R. Li, C.-T. Li, and Y. Guan, "Incremental update of feature extractor for camera identification," in *IEEE International Conference on Image Processing*, Quebec City, Canada, Sep. 27-30, 2015.

[33] R. Li, C. Kotropoulos, C.-T. Li, and Y. Guan, "Random subspace method for source camera identification," in *IEEE International Workshop on Machine Learning for Signal Processing*, 2015, pp. 1–5.

[34] J. Adams, K. Parulski, and K. Spaulding, "Color processing in digital cameras," *IEEE Micro*, vol. 18, no. 6, pp. 20–30, 1998.

[35] J. E .Greivenkamp, "Color dependent optical prefilter for the suppression of aliasing artifacts," *Applied Optics*, vol. 29, no. 5, pp. 676–684, 1990.

[36] S. Battiato, A. R. Bruna, G. Messina, and G. Puglisi, *Image processing for embedded devices*, Bentham Science Publishers, 2010.

[37] S. Bayram, *Applications of multimedia forensics*, Ph.D. thesis, Polytechnic Institute of New York University, New York, NY, 2012.

[38] G. E. Healey and R. Kondepudy, "Radiometric ccd camera calibration and noise estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 3, pp. 267–276, 1994.

[39] J. Lukas, J. Fridrich, and M. Goljan, "Determining digital image origin using sensor imperfections," in *SPIE Electronic Imaging, Image and Video Communication and Processing*. International Society for Optics and Photonics, 2005, pp. 249–260.

[40] M. Mhak, I. Kozintsev, and K. Ramchandran, "Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Mar. 1999, vol. 6, pp. 3253–3256.

[41] J. L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, Apr 1994.

[42] D. Donoho, M. R. Duncan, X. Huo, and O. Levi, "Wavelab 802 [online]," `http://statweb.stanford.edu/~wavelab/`.

[43] W. Liu, W. Zeng, L. Dong, and Q. Yao, "Efficient compression of encrypted grayscale images," *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 1097–1102, 2010.

[44] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[45] "BM3D matlab software," `http://www.cs.tut.fi/~foi/GCF-BM3D/`, Jan. 2014.

[46] M. Al-Ani, F. Khelifi, A. Lawgaly, and A. Bouridane, "A novel image filtering approach for sensor fingerprint estimation in source camera identification," in *IEEE International Conference on Advanced Video and Signal Based Surveillance*, Aug. 2015, pp. 1–5.

[47] A. Lawgaly, F. Khelifi, and A. Bouridane, "Image sharpening for efficient source camera identification based on sensor pattern noise estimation," in *Fourth International Conference on Emerging Security Technologies*, Sept. 2013, pp. 113–116.

[48] M. Badamchizadeh and A. Aghagolzadeh, "Comparative study of unsharp masking methods for image enhancement," in *Third International Conference on Image and Graphics*, 2004, pp. 27–30.

[49] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[50] E. J. Alles, Z. J. M. H. Geradts, and C. J. Veenman, "Source camera identification for low resolution heavily compressed images," in *IEEE International Conference on Computational Sciences and Its Applications*, 2008, pp. 557–567.

[51] E. J. Alles, Z. J. M. H. Geradts, and Cor J C. J. Veenman, "Source camera identification for heavily jpeg compressed low resolution still images," *Journal of forensic sciences*, vol. 54, no. 3, pp. 628–638, 2009.

[52] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, "Source digital camcorder identification using sensor photo response non-uniformity," in *SPIE Electronic*

Imaging, Security, Steganography, Watermarking of Multimedia Contents IX, 2007, vol. 6505, pp. 1G–1H.

[53] Y. Hu, B. Yu, and C. Jian, "Source camera identification using large components of sensor pattern noise," in *International Conference on Computer Science and its Applications*, 2009, pp. 291–294.

[54] X. Lin and C.-T. Li, "Preprocessing reference sensor pattern noise via spectrum equalization," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 1, pp. 126–140, 2016.

[55] C. R. Holt, "Two-channel likelihood detectors for arbitrary linear channel distortion," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 35, no. 3, pp. 267273, 1987.

[56] M. Goljan, "Digital camera identification from images - Estimating false acceptance probability," in *International Workshop Digital-forensics and Watermarking*, 2009, pp. 454–468.

[57] B.V.K. Vijaya Kumar and L. Hassebrook, "Performance measures for correlation filters," *Applied Optics*, vol. 29, no. 20, pp. 2997–3006, 1990.

[58] J. Fridrich and M. Goljan, "Derivation of ROCs for composite fingerprints and sequential trimming," Dept. Elect. Comput. Eng., Binghamton Univ., Binghamton, NY, Tech. Rep., Jan. 2010.

[59] M. Rosenblatt, "A central limit theorem and a strong mixing condition," *the National Academy of Sciences*, vol. 42, no. 1, pp. 43–47, 1956.

[60] J. Neyman and E. S. Pearson, *On the Problem of the Most Efficient Tests of Statistical Hpotheses*, NY: Springer, 1992.

[61] J. A. Swets, *Signal detection theory and ROC analysis in psychology and diagnostics: Collected papers*, Psychology Press, 2014.

[62] S. Bayram, H. T. Sencar, and N. Memon, "Efficient techniques for sensor fingerprint matching in large image and video databases," in *SPIE 7541, Media Forensics and Security II*, 2010, pp. 754109–754109.

[63] S. Bayram, H.T. Sencar, and N. Memon, "Sensor fingerprint identification through composite fingerprints and group testing," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 597–612, Mar. 2015.

[64] M. Goljan, J. Fridrich, and T. Filler, "Managing a large database of camera fingerprints," in *SPIE, Electronic Imaging, Media Forensics and Security XII*, San Jose, CA, Jan. 2010, vol. 7541, pp. 08.01–08.12.

[65] Y. Hu, B. Yu, and C. Jian, "Fast camera fingerprint search algorithm for source camera identification," in *International Conference on Computer Science and its Applications*, 2009, pp. 1–5.

[66] M. Goljan and J. Fridrich, "Sensor fingerprint digests for fast camera identification from geometrically distorted images," in *IS&T/SPIE Electronic Imaging*, 2013, pp. 86650B.1–10.

[67] Y. Hu, C.-T. Li, Z. Lai, and S. Zhang, "Fast camera fingerprint search algorithm for source camera identification," in *IEEE International Symposium on Communications Control and Signal Processing*, 2012, pp. 1–5.

[68] Y. Hu, C.-T. Li, and Z. Lai, "Fast source camera identification using matching signs between query and reference fingerprints," *Multimedia Tools and Applications*, vol. 74, no. 18, pp. 7405–7428, 2015.

[69] S. Bayram, H. Sencar, and N. Memon, "Efficient sensor fingerprint matching through fingerprint binarization," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 4, pp. 1404–1413, 2012.

[70] D. Valsesia, G. Coluccia, T. Bianchi, and E. Magli, "Compressed fingerprint matching and camera identification via random projections," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 1472–1485, Jul. 2015.

[71] D. Valsesia, G. Coluccia, T. Bianchi, and E. Magli, "Large-scale image retrieval based on compressed camera identification," *IEEE Transactions on Multimedia*, vol. 17, no. 1, pp. 1439–1449, Sept. 2015.

[72] D. Valsesia, G. Coluccia, T. Bianchi, and E. Magli, "Scale-robust compressive camera fingerprint matching with random projections," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 1697–1701.

[73] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," *Contemporary Mathematics*, vol. 26, pp. 189–206, 1984.

[74] M. Rajivkumar, B. V. Dhandra, and M. Gururaj, "Image recognition using texture and color," *IJCA National Conference on Recent Advances in Information Technology*, vol. 3, pp. 33–35, Feb. 2014.

[75] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, vol. 1, Elsevier, 2014.

[76] T. Gloe and R. Böhme, "The dresden image database for benchmarking digital image forensics," *Journal Digital Forensic Practice*, vol. 3, no. 2-4, pp. 150–159, 2010.

[77] C.-T. Li and R. Satta, "Empirical investigation into the correlation between vignetting effect and the quality of sensor pattern noise," *IET Computer Vision*, vol. 6, no. 6, pp. 560–566, 2012.

[78] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. New York: Academic, 1991.

[79] J. Ye, Q. Li, H. Xiong, H. Park, R. Janardan, and V. Kumar, "IDR/QR: an incremental dimension reduction algorithm via QR decomposition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 9, pp. 1208–1222, Sept. 2005.

[80] C. Yang, L. Wang, and J. Feng, "On feature extraction via kernels," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 2, pp. 553–557, April 2008.

[81] Y. Wang and K. N. Plataniotis, "An analysis of random projection for changeable and privacy-preserving biometric verification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 5, pp. 1280–1293, Oct 2010.

[82] B. Scholkopft and KR. Mullert, "Fisher discriminant analysis with kernels," *Neural Networks for Signal Processing IX*, vol. 1, pp. 1, 1999.

[83] X. Li, S. Lin, S. Yan, and D. Xu, "Discriminant locally linear embedding with high-order tensor data," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 2, pp. 342–352, April 2008.

[84] A. M. Martínez and A. Kak, "PCA versus LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228–233, 2001.

[85] G.H. Golub and C. F. Van Loan, *Matrix Computations*, Baltimore, MD: The Johns Hopkins University Press, 1989.

[86] E. Oja, "Subspace methods of pattern recognition," *Research Studies Press*, 1983.

[87] W. B. Johnson and J. Lindenstrauss, "On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix," *Journal of Mathematical Analysis and Application*, vol. 106, pp. 69–84, 1985.

[88] T.D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *IEEE Transactions on Neural Networks*, vol. 2, pp. 459–473, 1989.

[89] M. Artač, M. Jogan, and A. Leonardis, "Incremental PCA for on-line visual learning and recognition," in *IEEE International Conference on Pattern Recognition*, 2002, vol. 3, pp. 781–784.

[90] Y. Zhang and J. Weng, "Convergence analysis of complementary candid incremental principal component analysis," Dept. of Computer Science and Eng., Michigan State University, Tech. Rep. MSU-CSE-01-23, 2001.

[91] J. Weng, Y. Zhang, and W.S. Hwang, "Candid covariance-free incremental principal component analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 1034–1040, 2003.

[92] Y. Li, "On incremental and robust subspace learning," *Pattern recognition*, vol. 37, no. 7, pp. 1509–1518, 2004.

[93] D. Skocaj and A. Leonardis, "Weighted and robust incremental method for subspace learning," in *IEEE International Conference on Computer Vision*, 2003, pp. 1494–1501.

[94] Y. Guan, *Covariate-invariant gait recognition using random subspace method and its extensions*, Ph.D. thesis, University of Warwick, Coventry, UK, 2015.

[95] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998.

[96] R. Bryll, R. Gutierrez-Osuna, and F. Quek, "Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets," *Pattern Recognition*, vol. 36, no. 6, pp. 1291–1302, 2003.

[97] J. Kittler and F. Roli, "Multiple classifier systems," *Lecture Notes in Computer Science*, 2002.

[98] X. Wang and X. Tang, "Random sampling lda for face recognition," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004, vol. 2, pp. 259–265.

[99] N. V. Chawla and K. W. Bowyer, "Random subspaces and subsampling for 2-D face recognition," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, vol. 2, pp. 582–589.

[100] Y. Guan, C.-T. Li, and Y. Hu, "Random subspace method for gait recognition," in *IEEE International Conference on Multimedia and Expo Workshops*, Jul. 2012, pp. 284–289.

[101] Y. Guan, C.-T. Li, and S. D. Choudhury, "Robust gait recognition from extremely low frame-rate videos," in *IEEE International Workshop on Biometrics and Forensics*, 2013, pp. 1–4.

[102] Y. Guan, C.-T. Li, and F. Roli, "On reducing the effect of covariate factors in gait recognition: A classifier ensemble method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 7, pp. 1521–1528, 2015.

[103] X. Wang and X. Tang, "Random sampling for subspace face recognition," *International Journal of Computer Vision*, vol. 70, no. 1, pp. 91–104, 2006.

[104] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1991, pp. 586–591.

[105] L. I. Kuncheva, J. J. Rodríguez, C. O. Plumpton, D. E. J. Linden, and S. J. Johnston, "Random subspace ensembles for fmri classification," *IEEE Transactions on Medical Imaging*, vol. 29, no. 2, pp. 531–542, 2010.

[106] J. Marín, D. Vázquez, A. M. López, J. Amores, and L. I. Kuncheva, "Occlusion handling via random subspace classifiers for human detection," *IEEE Transactions on Cybernetics*, vol. 44, no. 3, pp. 342–354, 2014.

[107] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[108] J. Ye, "Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems," *Journal of Machine Learning Research*, pp. 483–502, 2005.

[109] J. Ye, R. Janardan, Q. Li, and H. Park, "Feature reduction via generalized uncorrelated linear discriminant analysis," *IEEE Transactionss on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1312–1322, 2006.

[110] A. C. Popescu and H. Farid, "Statistical tools for digital forensics," in *6th International Workshop on Information Hiding*, 2004, pp. 128–147.

[111] G. Shakhnarovich, L. Lee, and T. Darrell, "Integrated face and gait recognition from multiple views," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, vol. 1, pp. 439–446.

[112] Y. Guan, C.-T. Li, and Y. Hu, "An adaptive system for gait recognition in multi-view environments," in *ACM Multimedia and Security Workshop*, 2012, pp. 139–144.

[113] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen, "Multi-view discriminant analysis," in *European Conference on Computer Vision*, pp. 808–821. Springer, 2012.

[114] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen, "Multi-view discriminant analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 188–194, Jan 2016.