UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

ÉCOLE DOCTORALE SCIENCES DU NUMÉRIQUE ET DE L'INGÉNIEUR

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

Spécialité : TECHNOLOGIE DE L'INFORMATION ET DE LA COMMUNICATION

Certification de l'intégrité d'images numériques et de l'authenticité

---

présentée et soutenue par

Hoai Phuong NGUYEN

le 7 Décembre 2018

JURY

M. Jean-Luc DUGELAY	Professeur	Rapporteur
M. Patrick BAS	Directeur de recherche	Rapporteur
Mme. Christine FERNANDEZ-MALOIGNE	Professeur	Examinateur
M. Marc PIC	Docteur	Examinateur
Mme. Agnès DELAHAIES	Maître de conférence	Encadrante
M. Florent RETRAINT	Professeur	Directeur de thèse
M. Frédéric MORAIN-NICOLIER	Professeur	Directeur de thèse

UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

ÉCOLE DOCTORALE SCIENCES DU NUMÉRIQUE ET DE L'INGÉNIEUR

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

Spécialité : TECHNOLOGIE DE L'INFORMATION ET DE LA COMMUNICATION

Certification de l'intégrité d'images numériques et de l'authenticité

---

présentée et soutenue par

Hoai Phuong NGUYEN

le 7 Décembre 2018

JURY

M. Jean-Luc DUGELAY	Professeur	Rapporteur
M. Patrick BAS	Directeur de recherche	Rapporteur
Mme. Christine FERNANDEZ-MALOIGNE	Professeur	Examinateur
M. Marc PIC	Docteur	Examinateur
Mme. Agnès DELAHAIES	Maître de conférence	Encadrante
M. Florent RETRAINT	Professeur	Directeur de thèse
M. Frédéric MORAIN-NICOLIER	Professeur	Directeur de thèse

Acknowledgements

This work has been carried out within the Research Center of Information Technologies and Communication Systems (CReSTIC) and the Laboratory of Systems Modeling and Dependability (LM2S) at respectively the University of Reims Champagne-Ardenne (URCA) and the University of Technology of Troyes (UTT). It is conjointly funded by URCA, UTT and the group SURYS (Paris).

This work has been accomplished under the co-supervision of Professor Frédéric Morain-Nicolier (CReSTIC) and Professor Florent Retraint (LM2S). I would like to express my deepest appreciation to them and their highly professional guidance and incessant support. Both of them have accompanied with me from my master's intership at UTT and encouraged me to continue the research in the field. I highly appreciate the friendly and professional environment they have created for me during my three-years doctoral.

I would like to thank Mr. Mac Pic (SURYS) and Mrs. Agnès Delahaies for their supports and valuables advices during my doctoral.

I would like to express my special thanks to Mr. Jean-Luc Dugelay and Mr. Patrick Bas for accepting to review my PhD thesis. I also would like to address my gratitude to Mrs. Christine Fernandez-Maloigne and Mr. Marc Pic for agreeing to examine this thesis.

Most of all, I would like to thank my parents, my brother and my beloved Nana for always loving and supporting me.

Abstract

Nowadays, with the advent of computational technologies, imaging data in digital format can be easily and automatically processed with powerful computer systems, which permits us to design many useful applications, especially applications in the field of security. For applications using digital imaging data, depending on their purposes, there are two main problems: the falsification of imaged objects (input data) and the falsification of digital images (output data). The falsification of input data is the main vulnerability of applications such as people or product authentication. The falsification of output data is more sophisticated. Falsified images can be utilized as input for many applications such as police investigation, online marketing, political communication, etc. False information transmitted by these falsified medias can be very harmful for several entities. This thesis address these vulnerabilities for some specific digital imaging applications. We have studied and proposed solutions for detecting falsified input data from people and product authentication applications. The goal is to develop a statistical decision approach as reliable as possible that allows to guarantee a prescribed false alarm probability. To this end, the approach involves designing statistical tests within the framework of hypothesis testing theory based on a parametric model that characterizes physical and statistical properties of imaged objects in images. We have also studied and proposed an efficient solution for detecting digital image forgeries by exploiting the specific characteristics in the DCT domain of JPEG images.

Keywords: digital forensic, image processing, statistical test, image forgery detection

Résumé

De nos jours, avec l'avènement des technologies informatiques, il est possible de traiter facilement et automatiquement des données d'imagerie au format numérique à l'aide d'ordinateurs très performants. La capacité de traiter ce type de données avec des systèmes informatiques nous permet de concevoir une myriade d'applications intéressantes. Pour des applications exploitant des données d'imagerie numériques, dans le domaine de la sécurité, on est souvent confronté aux deux principaux problèmes suivants: la falsification des entrées du système d'imagerie (les objets imagés) et la falsification de la sortie du système (les images numériques). Le premier problème est la principale vulnérabilité des applications de sécurité telles que l'authentification des personnes ou des produits. La falsification des images numériques est plus inquiétante parce que des images falsifiées peuvent être utilisées dans divers domaines tels que les enquêtes policières, le marketing en ligne et la communication politique. L'information transmise par ces éléments falsifiés peut avoir des conséquences néfastes. Dans cette thèse, nous proposons des solutions permettant de détecter la falsification des objets imagés pour des systèmes d'authentification de personnes par reconnaissance faciale et pour des systèmes d'authentification de produits à partir de QR codes tatoués numériquement. La méthodologie proposée consiste à développer des approches de décision statistique aussi fiables que possible et permettant de garantir une probabilité de fausse alarme prescrite. Les approches proposées sont basées sur des modèles paramétriques caractérisant des propriétés physiques et statistiques des images numériques. Dans le cadre de ce travail, nous avons également développé une nouvelle solution pour la détection des falsifications d'images numériques en exploitant des caractéristiques dans le domaine DCT de la compression JPEG.

Mots clé: imagerie légale, traitement d'image, test statistique, détection de falsifications d'images

Contents

List of Figures

List of Tables

Chapter 1

General Introduction

Contents

1.1 General Context and Problem Description

Since the invention of the first camera, human-beings began to capture the real-world happening into images. Traditionally, images, which are captured through an analog acquisition device, are stored on photograph films. The manipulation of images over this physical support is extremely difficult. Imaging applications are therefore limited only to storage and artistic purposes. In the past few decades, the advent of digital imaging technology has radically changed the way we think about images. The passage from the analog to the digital world gives us a completely new and powerful way to treat image data. Digital data, which can be processed and manipulated computationally with the help of powerful computer systems, gives us the ability to think about more interesting imaging applications, such as: industrial inspection, medical imaging, traffic monitoring, 3D modeling & reconstruction, authentication of people or product, and so on. This thesis addressed some challenges related to specify digital imaging applications. The proposed works can be classified in the domain of digital forensics. Authentication systems using facial recognition are one of the digital imaging applications that we are interested in. These systems exploit facial images of a person as unique biometric characteristics to identify and then to authenticate this one based on the presence or absence of his previously saved extracted features. However, these systems are vulnerable against face presentation

attacks (a.k.a face spoofing attacks). In fact, an attacker can bypass the authentication process by simply presenting in front of the camera a copy version of the legitimate user's face image. With the growth of the internet and particularly of social networking, where many people don't mind publishing their personal information including their photographs, attacker can easily obtain the photos of somebody and use them finally to attack their owner. Therefore, it becomes crucial now to find out some ways to reducing or suppressing this vulnerability. Another application of digital imaging is on the certification of the authenticity and the originality of printed documents. In fact, forgeries related to printed documents can easily be performed with the aid of today's advanced electronic devices such as scanners and computers. The forged documents, which are usually undetectable by human eyes, can cause many harmful consequences. There is an important need to find solutions to the treat of counterfeiting such documents. The last problem that we have covered in this thesis is the image forgery detection problem. As mentioned above, digital images give us more rooms to develop many interesting applications. However, the ability to process more and more easily digital images with computer software brings us also some problems. The question is that if an image always gives you a correct information about the real world? We are gradually worrying about the integrity and the authenticity of digital images. In fact, with the emergence of photo-editing software, malicious people now can alter or modify the content of images without difficulty. They use the modified images to transmit wrong information to their public, which can lead to some negative consequences for others. In a very concrete case, if a falsified image is used as evidence in court, the judges probably make wrong judgments about the fact. Therefore, it is crucially important to develop image forensic techniques which can help us to certify the integrity and the authenticity of digital images.

1.2 Outline of the Thesis

The goal of this thesis is to investigate the different problems mentioned in the above section. The overall structure of the thesis is the following: Chapter 1 mainly introduces our research background and generally describes the problems studied in the thesis. It also presents in more details the outline of the thesis.

Chapter 2 provides an overview of current-art of related digital image forensics problems. It is also proposed to introduce the image processing pipeline which is the

fundamental for studying image forensics. Chapter 3 deals with the face presentation attack detection (PAD) problem. A new face PAD solution based on a statistical model of image noise is introduced. In fact, facial images from a presentation attack contains specific textural information caused by the presentation process which makes them different from face bona-fide images. The subtle difference between face bona-fide and presentation attack images can be interpreted by the difference regarding noise statistics. A noise local variance model is proposed and exploited to build statistical tests. We also introduce a new PAD database containing face bona-fide images and images of high-quality presentation attacks. The performance of the proposed solution is tested within the mentioned database. Experimental results show that, in controlled conditions, the proposed solution performs better than the other approaches in the literature.

Chapter 4 introduces a watermarking solution to secure printed QR (Quick Response) codes. The proposed method is developed in the context of the need for a cheap and efficient solution to protect product packaging against counterfeiting. It is proposed to embed a particular random micro-texture into the standard QR codes, which makes the printed codes more fragile to be reproduction and copy. Any degradations caused by the counterfeiting process change the statistical behaviors of the embedded micro-textures and can be detected. Statistical detectors based on the hypothesis testing framework are also introduced to classify authentic and counterfeited printed codes. Experimental results confirm the usability and the effectiveness of the proposed anti-counterfeiting solution.

In the chapter 5, a fast and efficient method is proposed to detect and localize tampered regions within falsified JPEG images. In fact, JPEG compression leaves specific traces of JPEG images. Tampered and untampered region may originate from images previously compressed under different configurations. This difference may introduce some inconsistencies between these regions. The inconsistency in the DCT domain is exploited in the proposed solution. The method is tested on some public image forgery detection databases and on a private one. The performance is promising.

The final chapter, chapter 6, concludes this thesis and present the perspectives of future work opened in the field of digital forensics.

1.3 Publications and Communications

Journal articles under review

1. H. P. Nguyen, A. Delahaies, F. Retraint, F. Morain-Nicolier, "Face presentation attack detection based on a statistical model of image noise",2018.

2. H. P. Nguyen, F. Retraint, F. Morain-Nicolier, A. Delahaies, "A watermarking technique to secure printed matrix barcode - Application for anti-counterfeit packaging",2018.

Conference papers

1. H. P. Nguyen, F. Retraint, F. Morain-Nicolier, A. Delahaies, "Face spoofing attack detection based on the behavior of noises", in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2016.

2. H. P. Nguyen, A. Delahaies, F. Retraint, D. H. NGuyen, M. Pic, F. Morain-Nicolier, "A watermarking technique to secure printed QR codes using a statistical test", in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2017.

3. K. T. Nguyen, C. Zitzmann, F. Retraint, A. Delahaies, F. Morain-Nicolier, H. P. Nguyen, "Face spoofing detection for smartphones using a 3D reconstruction and the motion sensors", in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 2018.

Conference papers under review

1. H. P. Nguyen, F. Retraint, F. Morain-Nicolier, A. Delahaies, "Tampering detection solution based on DCT coefficient analysis", 2018.

Chapter 2

Overview on Digital Image Forensics

Contents

2.1 Introduction

As discussed in the Chapter 1, we address in this thesis specific problems related to some digital imaging applications. In this chapter, we mainly provide an overview of the state-of-the-art elements of the three problems that we have approached. The section 2.2 introduces the image processing pipeline of digital cameras which permits us to understand the creation and the characteristics of digital images. Such understanding is useful for studying many problems related to digital imaging applications. The section 2.3 introduces the state-of-the-art solutions of the face presentation detection problem, which is studied in the chapter 3. In the section 2.4, we give an general view of printed document forensics that we have studied before finalizing the watermarking solution for protecting printed QR codes against reproductive cloning, introduced in the chapter 4. The third problem that we have studied in the thesis is the forgery detection in digital images. The section 2.5 describes different types of image forgeries and provides an overall view of the recent state-of-the-art in the field of image forgery detection.

2.2 Image Processing Pipeline in Digital Cameras

Nowadays, there are a myriad of imaging technologies which permit us to capture an image. However, this thesis only deals with digital images taken by Digital Still Cameras (DSC). These cameras come in many models, from cell phones, personal computers to complex professional equipment. They are the main sources which produce most of digital images in our days. Other sources of digitized images such as scanners or medical imaging systems are not addressed in this thesis. The image processing pipeline studied in this section refers therefore to the one of DSCs. Knowledge about the image processing pipeline is fundamental in this work. An overall understanding of the pipeline is required to be able to study efficiently digital images. The whole pipeline involves several operations from light capturing to image storage [1]. The sequence of operations differs from the manufacturer to the manufacturer but some basic operations remain similar. The whole pipeline is illustrated in the figure 2.1.

Incident photons radiating from imaging scene after passing through the optical system and the Color Filter Array are harvested and measured by the camera sensor, which produces an electrical signal. An analog-to-digital converter converts the

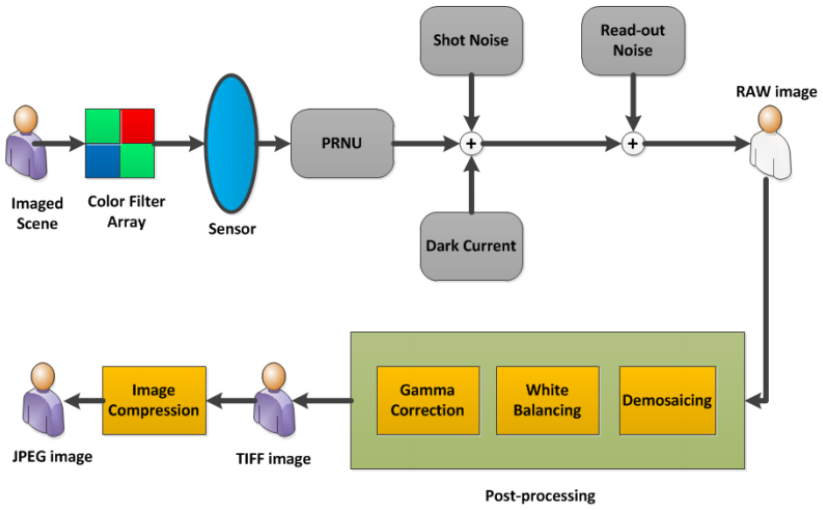


Figure 2.1: Illustration of the Image Acquisition Pipeline

electrical signal into a digital one. The digital signal containing information recorded by the image sensor is then stocked in a RAW image format. The RAW image goes through a set of typical post-acquisition process, e.g. demosaicing, white-balancing and gamma correction, to render a full-color high-quality image, referred as TIFF (Tagged Image File Format) image. The TIFF image file is often heavy and not very practical for transmission purposes. Therefore, an image compression process usually occurs at the final step to reduce the image size. The JPEG compression algorithm, studied in the thesis, is the most popular one because of its very good trade off between image size and quality. Full-color digital images are studied in this thesis. They can be represented under different color space but the most popular one is the RGB space. An RGB image consists of three color components: Red, Green and Blue. The combination of these three components is sufficient to represent millions of colors. Let define formally a full-color RGB image by a three-dimensional matrix of size $N_r \times N_c \times 3$ where N_r and N_c are respectively the number of rows and columns. Let denote $c \in \{R, G, B\}$ the color channel index. Typically, each pixel value is a natural integer. Denote v the number of bits using to decode a pixel, the set of possible pixel values is $\mathcal{Z} = \{0, 1, ..., B\}$ with $B = 2^v - 1$. To be clear for the next discussions, let $\mathbf{Z}$ denote an image in RAW format and $\mathbf{X}$ denote an image in TIFF or JPEG format. Each color component of the image $\mathbf{Z}$ is denoted by $\mathbf{Z}^c$ and a pixel of the color channel c at the location (m, n) is denoted by $z^c(m, n), 1 \leq m \leq N_r, 1 \leq n \leq N_c$

2.2.1 RAW Image Formation

A digital camera consists of an optical system, an image sensor and an electric system which can be compared to the pupil, the retina and the brain in the human visual system. The optical system consists of a set of optical lenses, which help to lead and focus incident light onto the surface of the image sensor. The image sensor consists of a two-dimensional array of photodiodes. Each one is in charge of collecting incident photons then producing an electrical signal which represents the light intensity of the corresponding image pixel. The output of the image sensor is processed by an A/D converter to obtain an initial digital signal, which is referred as the RAW image. Depending on the capacity of the A/D converter, a RAW image data can be coded with 12, 14 or even 16 bits. RAW images don't pass through any post-processing operations; they preserve all the original information of the captured scene given by the camera sensor.

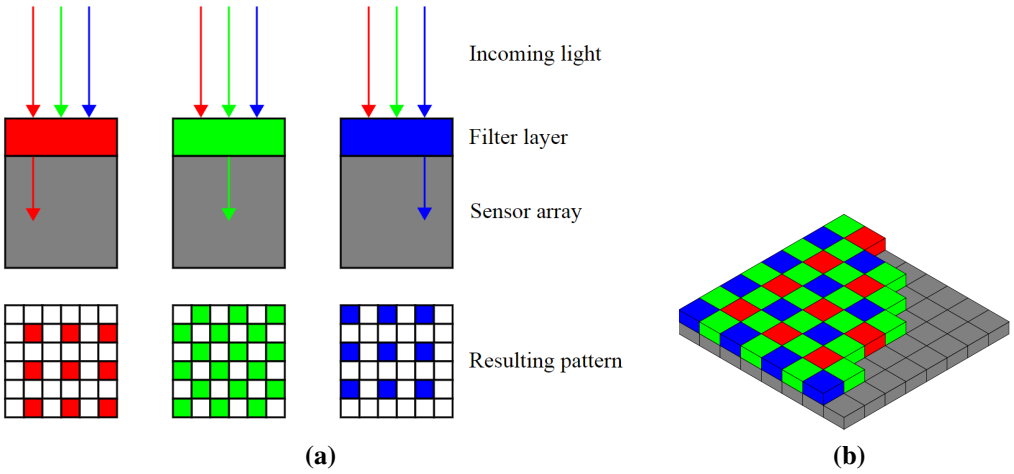


Figure 2.2: 2.2a Mechanism of CFA ; 2.2b The Bayer CFA mosaic. Each two-by-two submosaic contains 2 green, 1 blue, and 1 red filter, each filter covering one pixel sensor.

Photodiodes of image sensor are sensitive to luminous flux but they do not differentiate light wavelengths, which refers to the image color. It is proposed therefore to be overlaid on the image sensor a Color Filter Array (CFA), which is a mosaic of tiny color filters placed over each sensor's photodiode to limit only photons from a certain range of wavelengths, corresponding to either red, green or blue color, being able to reach the photodiode, Figure 2.2a. There are many types of CFA patterns and the most popular one is the Bayer CFA pattern, see Figure 2.2b. The human visual system is more sensitive with green light than the two other colors. Therefore, the Bayer CFA pattern contains twice as many green as red or blue samples. This

design, which based on the human visual system, enables to better capture the luminance component of light and, thus, provides better image quality. There are also some other image sensors, such as the *Foveon X3* sensor, which use another technique to capture color information. The *Foveon X3* sensor uses a two-dimensional array of photosites, each of which consists of three vertically stacked photodiodes. Each of these stacked photodiodes responses to different wavelengths of light, which penetrate the photosites' silicon to different depths. So that, full-color information can be recorded into a single pixel. However, due to the high production cost, these sensors are not used widely. We will not handle this case in this thesis. Let remind that we have denoted $\mathbf{Z}$ as the RAW image. Due to the CFA sampling, the obtained RAW image $\mathbf{Z}$ is then a single-channel image, which is represented by a two-dimensional matrix of size $N_r \times N_c$. Each pixel of the RAW image corresponds to the intensity of a certain color channel depending on its position. Each color component can be extracted from the RAW image by comparing with the CFA pattern grid. For a color channel c, its pixels value are given by

$$z^c(m,n) = \begin{cases} z(m,n) & \text{if } \mathbf{P}_{CFA}(m,n) = c \\ 0 & \text{otherwise,} \end{cases} \tag{2.1}$$

where $\mathbf{P}_{CFA}(m,n)$ represents the CFA pattern color at the position (m,n). To produce the RAW image, which is the very first digital version of the whole pipeline, it is unavoidable that several random noise sources are introduced:

- Shot noise, also referred as Poisson-distributed noise, which is related to the stochastic nature of the photo-counting process.
- Dark current noise generated by the thermal energy in the absence of light.
- Readout noise which is the noise of the on-chip amplifier which converts the charge into a change in analog voltage.

There is also a multiplicative noise source, referred as Photo Response Non-Uniformity (PRNU) noise, which is associated with the manufacturing of the image sensor. PRNU noise accounts for the difference of pixels response to the incident light caused by the imperfection and the non-uniformity of the photodiodes.

2.2.2 Post-Acquisition Processing

Due to the CFA sampling, the three color channels extracted from the RAW image are incomplete, Figure 2.2a. It is required to estimate all the missing color values at each pixel location in order to render a full color image. The missing values can be estimated from the value of pixels located within its neighborhood by interpolation. This operation is commonly referred to *CFA demosaicing*. There are many demosaicing algorithms, which can be generally categorized into *non-adaptive* and *adaptive* ones. For non-adaptive algorithms, a same interpolation technique is applied for all pixels.The nearest neighborhood, bilinear, bicubic, and smooth hue interpolations are typical example for interpolation technique used in non-adaptive solutions. All these interpolation techniques can be interpreted as a linear filtering operation. For example, the bilinear interpolation can be written as follows:

$$\mathbf{Z}^c_{DM} = \mathbf{H}^c_{DM} * \mathbf{Z}^c, \tag{2.2}$$

where $*$ denotes the two-dimensional convolution, $\mathbf{Z}^c_{DM}$ stands for the demosaiced image of the color channel c, and $\mathbf{H}^c_{DM}$ is the linear filter corresponding to the color channel c defined as follows:

$$\mathbf{H}^G_{DM} = \frac{1}{4}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{H}^R_{DM} = \mathbf{H}^B_{DM} = \frac{1}{4}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}. \tag{2.3}$$

Non-adaptive algorithms perform well in smooth regions of the image. However, they usually fail in textured regions and edges. Adaptive algorithms are therefore introduced to give a more accurate estimation of the full-color image. These algorithms employ edge information and/or inter-channel correlation to find an appropriate set of coefficients which permits to minimize the overall interpolation error. Adaptive approaches perform better than their non-adaptive counterpart. However, they are more computationally intensive. Anyway, all these demosaicing solutions lead to the linear correlation among adjacent pixels of images. After the demosaicing operation, we obtain a full color image. However, to improve the visual quality, some other post-processing operations are required [1]. We can mention here the two main operations: *white balancing* and *gamma correction*. White balancing is the process of removing unrealistic color casts, so that objects which appear white in person are rendered white in the image. In fact, due to the color temperature difference of light sources, a shift of the reflection spectrum of the object from its true color may occur.

This shift makes the object appear different in color when it is illuminated under different light sources. White balancing is basically performed by multiplying pixels in each color channel by a different gain factor. Denote g^c_{WB} the gain factor applied to the color channel c. The white-balanced image $\mathbf{Z}^c_{WB}$ is simply given by

$$\mathbf{Z}^c_{WB} = g^c_{WB} \cdot \mathbf{Z}^c_{DM} \tag{2.4}$$

The gain factors can be derived from the prior knowledge of the temperature information of the scene manually provided by photographers or automatically estimated by the camera system. Without this prior knowledge, some assumptions are used to estimate these factors. For instance, the classical white balancing algorithm, name as *Gray World*, assumes that the average value of three color channels will average to a common gray value

$$\bar{z}^R_{WB} = \bar{z}^G_{WB} = \bar{z}^B_{WB} \tag{2.5}$$

where $\bar{z}^c_{WB}$ denotes the average intensity of white-balanced image $\mathbf{Z}^c_{WB}$, given by

$$\bar{z}^c_{WB} = \frac{1}{N_r \cdot N_c} \sum_{m=1}^{N_r} \sum_{n=1}^{N_c} z^c_{WB}(m,n). \tag{2.6}$$

Denote similarly $\bar{z}^c_{DM}$ the average intensity of demosaiced image $\mathbf{Z}^c_{DM}$. According to Equation 2.4, g^c_{WB} values have to satisfy the following equations

$$g^R_{WB} \cdot \bar{z}^R_{DM} = g^G_{WB} \cdot \bar{z}^G_{DM} = g^B_{WB} \cdot \bar{z}^B_{DM} \tag{2.7}$$

The Gray World algorithm proposes to choose the green channel as reference due to the sensitiveness of the human visual system with this channel by assigning $g^G_{WB} = 1$. The gain factor of the other color channels is therefore finally given by:

$$g^R_{WB} = \frac{\bar{z}^G_{DM}}{\bar{z}^R_{DM}}, \quad g^B_{WB} = \frac{\bar{z}^G_{DM}}{\bar{z}^B_{DM}}. \tag{2.8}$$

At this stage, the pixel intensity is generally still linear with respect to RAW image intensity [1, 2]. However, the transfer function of most display devices, particularly the cathode ray tube (CRT) monitor, can be approximated by a exponential relation between the luminance L and the voltage V

$$L = V^\gamma \tag{2.9}$$

where γ is a constant specific for each device. Typically, $\gamma = 2.2$ for the CRT monitor. To compensate for this non-linear characteristic of display device and render the luminance into a perceptually uniform domain, the gamma correction process is required. It is pixel-wise non-linear operation, which is simply the inverse of the Equation 2.9, defined as follows:

$$z^c_{GC}(m,m) = (z^c_{WB}(m,m))^{\frac{1}{\gamma}}. \tag{2.10}$$

Furthermore, human eyes are much more sensitive to changes in dark tones than they are to similar changes in bright tones. The gamma correction process also permits to encode and store tones more efficiently. After all these post-processing operations, we obtain finally a full-color high-quality image, referred as TIFF image. In the following sections, we denote the final rendered TIFF image as $\mathbf{X}_{TIFF}$.

2.2.3 Image Compression

The high-quality TIFF image obtained from the previous operation is not adapted for transmission or storage purposes due to its very large size. Therefore, the last operation required by most of camera systems is the compression process, which usually employs a lossy compression algorithm, to reduce the data size. Lossy compression algorithms, as their name mentions, remove some image data, which may be considered as less sensible to the human visual system, to favor the data-size reduction. The lossy compression is then irreversible. Images reconstructed from the compressed ones are not identical as their original TIFF image. There is always a trade-off between the storage size and the visual quality of compressed images. When a high compression factor is applied, compressed image requires less space for storage, but the visual quality of the reconstructed image will also be gradually degraded. Many lossy compression algorithms for image data have been proposed. However, JPEG compression [3], which can remarkably strike the balance between image visual quality and size is the one utilized predominately by most of manufacturers. It is also the most popular compression technique used for image transmission over the internet. Each manufacturer can propose their own design for the compression scheme based on the standard one [3] to optimize the trade-off mentioned previously by dealing with three principal settings: color space on which the algorithm is applied, sub-sampling technique and quantization tables. The fundamental steps of JPEG compression and decompression process are illustrated in Figure 2.3. The compression technique proposed in JPEG is inter-channel independent.

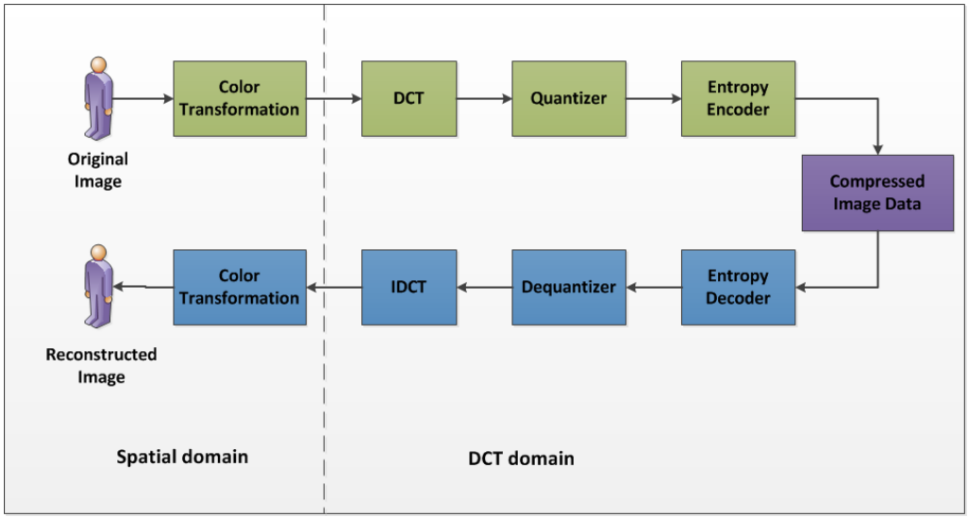


Figure 2.3: JPEG compression and decompression scheme.

In fact, each color channel is compressed separately. JPEG compression can be applied directly on the image under RGB color space, but it works better under YCbCr color space. There are usually strong correlations among the three channels of RGB image. YCbCr is introduced to reduce this inter-channel correlations. Y channel stands for the luminance intensity, Cb and Cr channels represent respectively the blue-difference and red-difference chroma components. These two chroma components usually contain less high frequency information than their RGB counterparts, so that JPEG compression works more efficiently in these components. Furthermore, human visual system is more sensitive with the luminance than the chroma. By separating these components, we can opt to privilege more space to save the luminance by down-sampling the two chroma channels by a factor of 2 horizontally and vertically. YCbCr color spaces are defined by a mathematical coordinate transformation from an associated RGB color space, given by

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & 0.081 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix}. \tag{2.11}$$

Each color component is then divided into non-overlapping 8×8 blocks. The compression step, which consists of Discrete Cosine Transformation (DCT) and quantization, is then applied independently on each 8×8 block. The DCT operation converts pixel values from spatial domain into a frequency one represented by a set

of DCT coefficients

$$I(u,v) = \frac{1}{4} T_u T_v \sum_{m=0}^{7} \sum_{n=0}^{7} x(m,n) \times cos\left(\frac{(2m+1)u\pi}{16}\right) cos\left(\frac{(2n+1)v\pi}{16}\right) \quad (2.12)$$

where $x(m,n)$ is a pixel in a 8×8 block, $I(u,v)$ denotes the DCT coefficient located the position $(u,v), 0 \leq u,v \leq 7$, and T_u are constant defined as follows:

$$T_u = \begin{cases} \frac{1}{\sqrt{2}} & \text{for} \quad u = 0 \\ 1 & \text{for} \quad u > 0 \end{cases} . \quad (2.13)$$

The DCT transform is reversible, its inverse operation is the IDCT transform. So that, the DCT transform is a lossless operation. The lossy process of JPEG compression occurs in the following step: *DCT quantization*. In DCT domain, we obtain 64 coefficients. The coefficient located at position $(0,0)$, referred as Direct Current (DC) coefficient, represent the average value of pixels within the associated block. The other 63 coefficients, referred as Alternating Current (AC) coefficients, represent information in different frequencies. For natural image, most of image energy is represented by the low-frequency components (i.e. the upper left corner of the 8×8 grid). High-frequency components represent detail information in spatial domain, which is less significant to human visual system. The quantization process permits to eliminate somewhat high-frequency components. It is performed by simply dividing each DCT coefficient by the corresponding quantization step, and then rounding to the nearest integer. This operation is formulated as follows:

$$D(u,v) = \texttt{round}\left(\frac{I(u,v)}{\Delta(u,v)}\right) \quad (2.14)$$

where $D(u,v)$ denotes the quantized DCT coefficient and $\Delta(u,v)$ the corresponding quantization step, an element of the quantization table. The configuration of the quantization table depends on the expected quality of the reconstructed image. The DCT quantization operation is irreversible. Quantized DCT blocks contain mainly a lot of zero-values in high-frequency components. By arranging quantized DCT coefficients into zig-zag path, we obtain usually sequences with a lot of contiguous zero-values. Run-Length Encoding (RLE) and Huffman coding are then applied to reduce a little bit more the size needed to encode the quantized image. This operation, referred as entropy coding, is perfectly irreversible. The JPEG decompression process

simply works in the reverse order, but with the inverse operation of the one employed in the compression process: entropy decoding, dequantization, IDCT transform, and the color space inverse transform (optional). First of all, the entropy decoding is performed to obtain perfectly the quantized image. Then, the dequantization is applied by simply multiplying the quantized DCT coefficient with the corresponding quantization step. Denote $I_{IDCT}(u,v)$ the dequantized DCT coefficients within a 8×8 block, we have

$$I_{IDCT}(u,v) = D(u,v) \cdot \Delta(u,v). \tag{2.15}$$

We return after that to the spatial domain by applying the IDCT transform on the dequantized DCT coefficients. Denote $x_d(m,n)$ a pixel value within a block obtained after the IDCT operation, so that

$$x_{IDCT}(m,n) = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} T_u T_v I_{IDCT}(u,v) \times cos\left(\frac{(2m+1)u\pi}{16}\right) cos\left(\frac{(2n+1)v\pi}{16}\right). \tag{2.16}$$

Typically, the x_{IDCT} values are not integer and may exceed the finite dynamic range (e.g. $[0,255]$). The final reconstructed value, denoted as x_{JPEG}, is therefore obtained by rounding and then truncating the latter:

$$x_{JPEG} = \texttt{trunc}\left(\texttt{round}\left(x_{IDCT}\right)\right). \tag{2.17}$$

If an YCbCr image is obtained after the last operation, a transformation into the RGB color space is then required to finalize the decompression process. The reconstructed JPEG image differs from the original TIFF one due to the information loss caused by quantization process and errors caused by the rounding and truncation operations, which appear several times in the whole pipeline.

2.3 Overview on Face Presentation Attack Detection Problems

2.3.1 Introduction

Automated face recognition is now widely used in many application areas ranging from access control and surveillance to commerce and entertainment [4, 5]. The

widespread use of these applications raises new security concerns, in which the vulnerability against presentation attacks attires the most attention. Face presentation attacks, also referred as the face spoofing attacks, are performed by presenting falsified data in front of the acquisition sensors using different Presentation Attack Instruments (PAIs). Basing on the type of PAI, face presentation attacks can be categorized into three types:

- *Print-Attack:* it is the simplest one to be implemented. Attackers use just a single image of a legitimate user's face printed on paper to bypass the authentication system. The printed face can also be wrapped to simulate the human face curvature.
- *LCD-Attack:* an image or a video capturing the legitimate user's face is displayed on a LCD screen of a portable device.
- *Mask-Attack:* it is more sophisticated. Attackers have to collect enough information about the legitimate user's face to create a 3D mask, which will be used after that to trick the system.

Existing approaches for face presentation attack detection are quite numerous. Many ways for categorizing these approaches have been proposed in literature [6, 7, 8]. A very first categorization, proposed by Chakka et al.[6], divided existing solutions into three categories: motion, liveness and texture analyses. This categorization is ambiguous and does not help to identify the strength of each solution. Furthermore, many recent approaches cannot be appropriately assigned by using this one. Tirunagari et al. [7] proposed to classify a method as either data-driven or cue-based. While cue-based approaches rely on intuition and observations, data-driven ones tend to use image-processing algorithms to extract some texture descriptors and then use them as data to classify images. Recently, Boulkenafet et al. [8] proposed a three-part categorization: hardware-based, challenge-response and software-based techniques. In this work, based on the type of input data, we propose a hierarchical categorization for face presentation attack detection solutions as given in Figure 2.4. A standard camera sensor is the basic source of information for most of the existing approaches, but alternative or additional information from other sensors can also be exploited in some approaches. Therefore, we propose to divide existing solutions into two principal categories: *hardware-based* and *software-based* approaches.

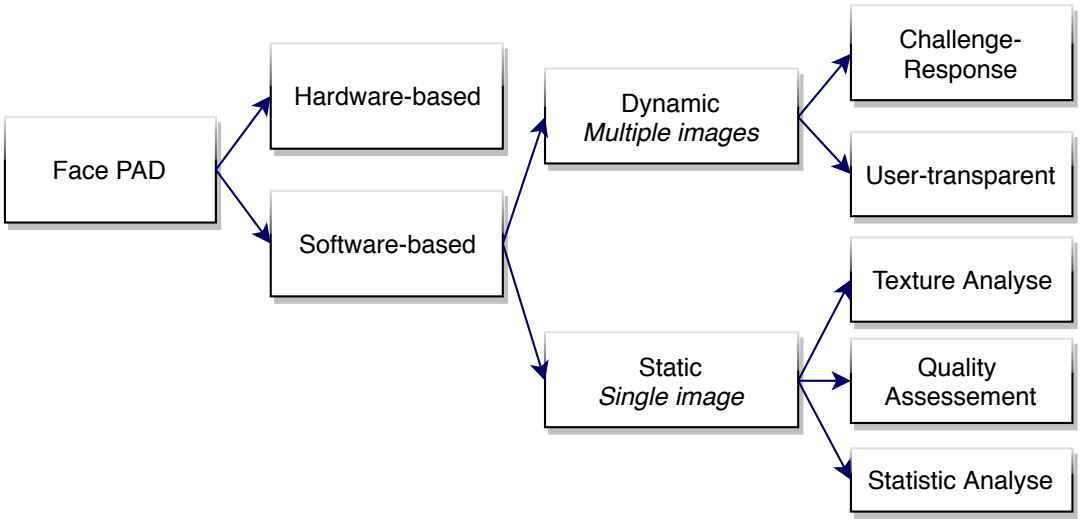


Figure 2.4: Proposed categorization of face PAD solutions.

2.3.2 Hardware-based face PAD solutions

Conventional camera sensor is the most basic and the most available source of data providing us information exploitable for dealing with face PAD. However, some other types of sensor can also give us valuable information. Hardware-based solutions exploit information from these sensors to provide efficient means for face presentation attack detection. Some particular types of camera sensor such as depth camera sensors [9, 10], stereo cameras [11, 12], near-infrared cameras [13, 14, 15], thermal cameras [16, 15], light field cameras [17, 18] and multi-spectral cameras [19] have been exploited in literature for detecting face presentation attacks.

Movement information of the camera device, given by motion sensors, can also be compared with the ones estimated from a 3D reconstruction process [20] for detecting presentation attacks. In a genuine authentication case, the two sources tend to describe the same information, which is not the case in a face presentation attack. The correlation between information given by the two sources can be used as a cue to detect presentation attacks. Hardware-based approaches require the use of alternative or additional sensors, which implies the restriction of their application.

2.3.3 Software-based face PAD solutions

Software-based approaches exploit only information given by conventional camera sensor. Based on the quantity of information exploited, we can divide these approaches into two sub-categories: *Dynamic* approaches which require multiple images and *static* ones where a single image is enough to take a decision.

Dynamic approaches

Approaches from this subcategory tend to detect face liveness or some specific 3D properties of the scene based on some a priori assumptions. This information can be interpreted as some specific dynamics between successive images, which cannot be extracted from a single image. Dynamic approaches can be categorized into *challenge-response* or *user-transparent*. A challenge-response approach requires user's cooperation. Authentication system prompts a user for some specific random actions, such as changing face expressions [21, 22], changing head pose [23, 24, 25], winking eyes [26], moving the pupils [27, 28], or speaking some specific words [29]. User's actions is probably absent or non-correlated with the required actions in the case of a face presentation attack. These approaches are usually time-consuming. They can also make users feel uncomfortable. Meanwhile, user-transparent approaches do not require user's cooperation. Spontaneous physiological signs of life such as facial visual dynamics [7, 30], eye-blinking [31] can be detected and used as liveness cues for face presentation attack detection. Some other approaches assume and exploit the significant difference between a bona-fide face, which is a complex non-rigid 3D object, with a planar object in the case of Print and Replay attacks: the difference in the optical flow field [32, 33], the variation of pixel values between images taken in different focuses [34, 35]. Different types of dynamic texture descriptor such as LDP-TOP [36], LBP-TOP, LPQ-TOP, BSIF-TOP [37], LBP-MDCT [38] representing both spatial and temporal dimension have been also proposed for face PAD. Some authors [39, 40] assumed that the overall motion of the face and the background is highly correlated in the case of presentation attack and inversely in a bone-fide case. However, this assumption is not correct in the case of Replay and Mask attacks. By cutting out tightly the background from a printed photo of a legitimate user, we can also produce a presentation attack that cannot be detected.

Static approaches

Single image requirement makes these approaches transparent and therefore non-intrusive. These approaches assumed that the inherent disparities between a bona-fide face and a presentation attack one could be observed from a single image. Two approaches have been introduced in literature: *Image quality assessment* and *Texture Analysis*. The mentioned inherent disparities may be interpreted as the difference in the image quality. Different image quality assessments can be used to interpret

this difference. Galbally et al. [41] proposed to use 14 image quality features extracted from a single image. The performance of this approach is highly dependent on the quality of the acquisition process. Bhogal et al. [42] proposed a set of 6 non-reference image quality measures to distinguish real biometric data from data as used in presentation attacks. Nikisins et al. [43] proposed a system built of different Image Quality Measures (IQM) forming a feature space, and a Gaussian Mixture Model which is trained to represent the probability distribution of bona-fide sample. The authors claimed that the system can detect unseen presentation acttacks. Texture analysis based approaches assumed that the nature of texture patterns presenting on face bona-fide images and fake ones should be different because of the imperfections and degradations caused by the presentation process. There are many things which can appear to make a presentation attack image different from a bona-fide one, such as light reflectance, noise, periodic patterns caused by the halftone process in a printed image or by discretization process in an image displayed on a screen, etc. The authors in [44], [45] proposed to study these differences in the frequency domain. Some authors proposed to analyze in spatial domain by using a local texture descriptors, such as Local Binary Pattern (LBP) [46, 47, 48], Local Graph Structure (LGS) [49, 50]. Different from other authors who study only the gray image, [8] proposed to exploit the texture-information in color spaces. These approaches usually employ a non-parametric classification solution such as Support Vector Machines (SVM) to help them classify images. Presentation attacks' images are needed for training the classification system. The classification performance depends therefore significantly on the quality and the size of the training dataset. It may significantly fail when we use the trained classification system to predict a presentation attack which is not present in the training database. In the chapter 3, we propose a novel static approach basing on a statistical analysis of noise. In fact, considering that the inherent disparities between a bona-fide and a presentation attack images can be interpreted as the difference in the behavior of noise in the image, we develop a statistical noise model which takes into account this difference. We use this model to build a face presentation attack detector with the help of the hypothesis testing framework. The proposed solution requires only a single image to process. Different from most of the related methods in the literature, the proposed solution uses a parametric approach. Once the model of face bona-fide images is constructed, there is no need to train the detector with face presentation attack images. The performance of the proposed solution can still be held when a new type of presentation attack is considered.

2.4 Overview on Printed Document Forensics

2.4.1 Introduction

With the emergence of technology, printing has become fast, low-cost and high quality. Printed objects, such as documents, tickets, contracts, certificates, patient records, bills, product labels or packages, etc. present widely and play an important role in many situations. These objects are used to exchange some sort of information between different entities such as government departments, companies and individuals. However, these objects can be easily forged or altered using physical or chemical manipulations, or sophisticated software. Depending on the information exchanged, the forgery of the alteration of these objects may cause different severe consequences, which are related to the violation of the authenticity or of the integrity of the information exchanged. For example, altered documents have been used increasingly for transaction and communication resulting in economic crimes and disturbance of social order. Counterfeited labels and packages help introduce fake products into the market, which damage the economic system, harm legal companies, and affect directly consumer health and safety. In the process of criminal investigation or fighting against counterfeiting, it is important to be able to track the sources of these printed objects, to understand their formation history, and to verify their authenticity and integrity. Product labels and packaging need to be protected. Printed documents, which are used as evidences in the tribunal, need to be examined. There are two families of approaches which can be used to resolve the security problems of printed objects, and to protect their authenticity and integrity: *passive forensics* and *proactive security measures*. Passive forensics aims to trace the sources of the objects basing on their intrinsic fingerprints. Proactive approaches aim to embed in the printed objects some watermarked information, which is served as a security element. A brief state-of-the-art of these approaches are given respectively in the section 2.4.4 and 2.4.5. In this study, we focused on print forensics techniques which are based on digitalized images of the printed objects. Approaches, which do not base on these elements, such as florescent inspection, microscopic examination, chemical inspection, etc. will not be covered here. In fact, these approaches often require special and expensive inspection equipment, and professional testing personnel. They also extend beyond the scope of our study. Approaches based on digitalized images permit to reduce the reliance on equipment and personnel, to improve detection efficiency as well as reduce testing costs. Although studied print forensics are also based on

digital images, they differ greatly from digital image forensics. The difference may be in regards to image sources, image contents and also the tampering techniques being detected.

- Digital image forensics work with images captured by camera, which are then stored on hardware and cannot be degraded over time. Images used for print forensics are digitalized from printed objects, which age over time and can be easily damaged or polluted during their circulation and storage. Before images of the printed objects are captured, we have already probably lost some valuable information.
- Digital images have rich content, containing lots of texture and color information. Meanwhile, images of printed objects, such as text documents, may often be monotonous and lack of textural information.
- The tampering of printed objects occurred before the digitalizing process. We cannot study print forensics based only on the characteristics of digitalized images themselves like in the case of digital image forensics. We have to consider eventually, in the study, the signatures of printed objects introduced during the printing process and the tampering one.

From an electronic document, two essential processes occur engagingly and successively to obtain the final digital image that we use to study for print forensics: printing and digitalizing. Printed documents can be created using different printing techniques. The most popular ones are laser printing and ink-jet printing. There are also different ways to create a digital image from printed documents. We can use a flatbed scanner as well as a portable camera system such as smartphones, cameras. Anyway, each combination of these processes creates and leaves their own signatures on the studied image. Hence, the two processes cannot be studied separately. The term *Print-and-Scan* process is then preferred to represent the whole process which transforms an electronic document into a digital image though a print and scan system. The process will be discussed more in detail in the Section 2.4.2.

2.4.2 Printing and scanning process

Printing process

The goal of printing operation is to transform the document into dots on a piece of paper. The printing document is first transformed into a print-ready halftone image,

which can be interpretable by the print engine. The transformation consists of two principal steps:

- Rendering the initial document file into a bit-mapped form. At the output stage of this step, an 8-bits/channel CMYK or RGB continuous tone document image is generated.
- The continuous-tone document is then subsequently modified by color space conversion, gamut mapping and finally halftoning.

Based on the entity which process the transformation, there are two different types of printing system: *host-based printing* and *controller-based printing*. Host-based printing refers to printing systems which use the host computer to process the transformation. Meanwhile, controller-based printing refers to printing systems which process the transformation directly by the printer controller. The halftoned document image is then processed by the print engine to produce the hardcopy document. Basing on the type of print engine, we consider here two types of printer, which dominate the market: *Laser Printer* and *Ink-jet Printer*. *Laser printer* systems use

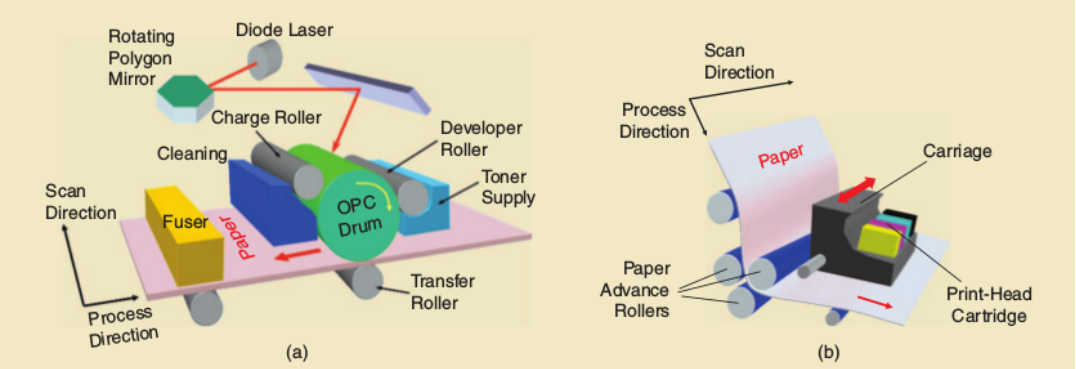


Figure 2.5: Architectures of (a) typical laser and (b) ink-jet printers.

the electrophotographic (EP) technology. A typical EP process consists of six steps, Figure 2.5 (a):

- **Charging-** Applying a negative charge to an organic photoconductive (OPC) drum, which rotates at a constant angular velocity, through a charger roller.
- **Exposing-** A laser beam reflected by a rapidly spinning polygon mirror is employed to neutralize selectively the charge on the surface of the OPC drum, leaving an electrostatic negative image of the printing document on the OPC drum's surface.

- **Developing-** A toner image is developed by electrostatically adhering toner particles to the discharged areas of the OPC drum's surface. Toner particles are transferred from the toner supply to the OPC drum via the developer roller.
- **Transferring-** The output media, typically a sheet of paper, is rolled under the OPC drum. The developed image consisting of toner particles will be transferred from the OPC drum's surface to the output media through a charged transfer roller.
- **Fusing-**The toner image is fused to the media using heat and pressure by the fuser.
- **Cleaning-** To finish and to prepare for the next printing, the OPC drum is cleaned by a cleaning blade that scrapes across the OPC drum's surface. All residual toner will be removed.

The laser printers generates inherently different artifacts due to the imperfection of their physical components such as the gear mechanism, polygon mirror wobble, OPC angular velocity, etc. These artifacts can be considered as intrinsic signatures of the device [51]. For example, the fluctuations of OPC angular velocity and errors in the gear transmission mechanism can produce banding artifacts, which appears as nonuniform and parallel light and dark lines. Depending brand and model, different laser printers have different sets of banding frequencies. In addition to the banding artifact, there are also geometric distortions of pages caused by spinning velocity fluctuations of the polygon mirror and imperfections of the paper feeding mechanism. These distortions also differ between different brands of laser printer. *Ink-jet printer* system consists of three principal components, Figure 2.5 [51] (b): *the printhead*, *the carriage*, and *the paper-advance mechanism*. The paper is picked up and advanced in the process direction under the carriage by the paper-advance mechanism. The carriage carrying the printhead moves back and forth across the paper in the scan direction, which is perpendicular to the process direction. The printhead is programed to fire synchronously drops of ink onto the paper during the moving of the carriage. The printhead consists of an array of nozzles which are connected with the ink reservoir. Figure 2.6 [51] illustrates the process of printing a three-pixel-wide vertical line using the typical two-pass print mode. The print mask consisting of a 2-D array of 1s and 2s is tiled over the whole page. According to the label of pixels from the print mask, the printhead will fire in the pass one (right to left) or pass two (left to right). Each column of nozzles allows to print out simultaneously several

rows of pixels during a single pass of the printhead across the paper. When the printhead has completed an entire pass in the scan direction, the paper is advanced in the process direction and the printhead prepares for a new pass. The process is repeated until the entire page is printed out and ejected into the output tray.

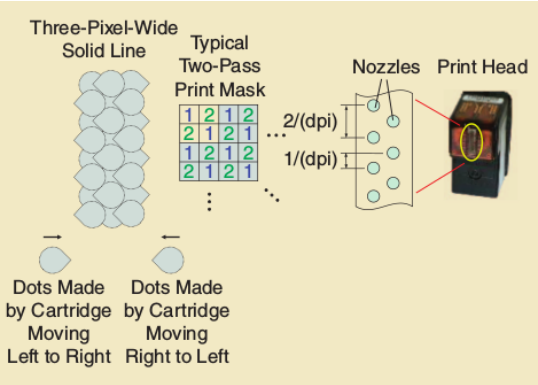


Figure 2.6: Ink-jet printing - Illustration of the process of printing a three-pixel-wide vertical line in a two-pass, bidirectional print mode.

Because the moving of the printhead and the firing of ink are synchronous, the shape of ink dots is not symmetrical. Each ink drop will have a small tail, Figure 2.6. Depending on the printing speed and direction, the tail can have different forms. At high printing speed, the tail may break away the main ink-drop to form a satellite, Figure 2.7 [51]. Hence, the appearance of the ink-drop can be used as an intrinsic signature of ink-jet printers. Another ink-jet print artifact is dot placement errors, which can be caused by paper advance errors, carriage positioning errors, or misaligned nozzles in the printhead. These errors are potentially useful signatures for forensics purposes.

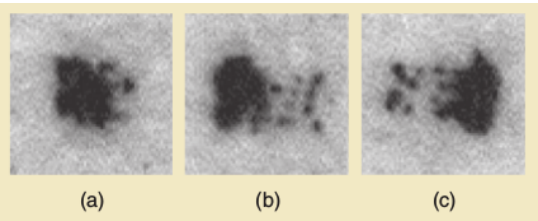


Figure 2.7: Ink-jet printing - Typical dots printed by a 300-dpi inkjet printer: (a) 15 inches/ second left-to-right print mode (b) 45 inches/second left-to right print mode, and (c) 45 inches/second right-to-left print mode.

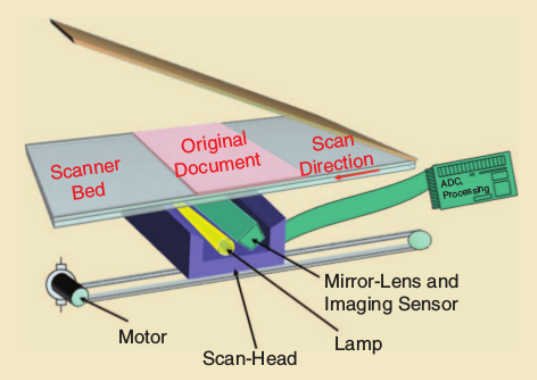


Figure 2.8: Typical architecture of a flatbed scanner.

Scanning process

The scanning process permits to transform a hardcopy document into a digital one. Different types of document scanner have been invented, but flatbed scanners are actually the ones which dominate the market and which produces high quality images. Faltbed scanners work by shinning white light onto the object to be scanned and then interpreting reflective light into intensity and color information. A flatbed scanner comprises typically a scanner bed, motor, scan head, lamp, mirror lens and imaging sensor, as shown in Figure 2.8. Most of faltbed scanners use a 1-dimension CCD/CMOS imaging sensor. The scanning process begins by putting a document on the scanner bed. The lamp illuminates the document through the scanner bed and lights will be reflected onto the imaging sensor. The imaging sensor, moved by the motor, scans the document line by line until finishing. The resolution of a flatbed scanner is determined by the number of elements in the linear imaging sensor and the step size of the motor controlling the scan head.

Digital cameras can be used for the same purposes as scanners. However, in comparison to a scanner, a camera image is subject to a degree of distortion, reflections, shadows, low contrast and blur. For most of flatbed scanners, data is captured by tri-linear CCD sensors. We don't have to process a democaicing operation to obtain a full color image, but these sensors introduce inevitably noise into the scanned image. These noises can be exploited as fingerprints of the related scanner for forensics purposes. There are three types of noise from sensor: sensor array defects, sensor pattern noise and sensor random noise. Array defects is related to local defections of the sensors' array: hotpoint defects, dead pixels, pixel traps... These defects cause pixels values in the scanned image to deviate greatly. Noise caused by these defects is easily corrected in most of the devices available in today's market. Pattern noise is

caused by dark current and the non-uniformity of photo response in the sensor array (PRNU noise). The PRNU noise is independent of image content. Random noise may be referred to photons noise which is strongly correlated with image content. Being independent to image content, the sensor array defects and the PRNU noise are often considered for source identification problems. Dust and scratches persisting on the surface of the scanner bed affect directly on the light beam irradiating the document. These can also, by consequence, cause some kind of distortion in the scanned image. Dust and scratches may present for a long time. Distortions caused by these can be considered as a fingerprint of a given scanner.

Print-and-Scan process

Printing followed by scanning involves conversion from digital to analog, and then back to digital form. The whole process is inherently very complex. The problem is amplified by the fact that a variety of printing and scanning devices are available in the market, which works on one of many different existing technologies. Modeling completely the print-and-scan process is then very difficult. By focusing on inkjet/laser printers and flatbed scanners, some authors have tried to propose a model the print-and-scan process. Lin and Chang [52] presented a very first model for the print-and-scan process, by considering pixel value distortions and geometric distortion. The distortions of pixel values are caused by variations of luminance, contrast, gamma correction, and chrominance, and the blurring of adjacent pixels. Geometric distortion is caused by rotation, scaling, and cropping. The authors focused on the changes of DFT coefficients for invariant extractions. Solanki et al. [53] proposed a model for the print-and-scan process by breaking it down into simpler subprocesses. The authors introduced different subprocesses that cause distortions during the print-and-scan process. Before printing, in order to make sure the printed images appear the same as on a monitor, a nonlinear operation, called *gamma tweaking*, is applied. After that, the image is converted to a *digital halftone* before it is printed. During printing, there are the *dot gain* and *print-to-print instability* distortions. Dot gain refers to the phenomenon that the printed images tend to appear darker than expected values due to several physical reasons, such as the spreading of ink on the medium. Print-to-print instability, according to the authors, is related to corretaled noise caused by uncertainties during the printing process. Banding distortion is shown as an example for Print-to-print instability. During the scanning process, the authors invoked three processes: *scanner gamma compensation*, *digitization*, and

geometric transformations. Scanner gamma compensation is similar to the gamma correction operation of the image processing pipeline of digital cameras described in the previous section. The operation permits to display correctly the scanned image on a monitor. The authors also exploited the DFT domain to analyze the effect of the print-and-scan process.

2.4.3 Forgeries of printed and scanned Documents

Forgeries of printed and scanned documents are malicious operations which tend to modify the information transmit by the document. For printed documents, forgeries occur after printing. Forensic investigators have directly on their hand the printed document and can examine it physically or chemically. If a digital approach is used, they have to scan the document themselves. In anyway, they have control of the scanning process. We can mention here some methods of printed document forgeries:

- Adding extra information into a given printed document by superposing a new printing on it.
- Erasing physically or chemically some information from a given document.
- Two part of a document are physically combined together, then the combined document is photocopied to create an integral forgery.

For scanned documents, investigators may not have access to the printed version. In general, they don't have the control of the scanning process. Therefore, forgeries can occur both before and after the scanning process. Image processing softwares, such as Photoshop or GIMP, can be used by malicious people to modify digitally the scanned document.

2.4.4 Passive approaches

As mentioned previously, passive approaches seek forensics solutions based on the intrinsic fingerprints of the printing and scanning systems. Ali et al [54] studied the banding artifact of laser printers. The authors claimed that the banding frequency among the units of the same model of laser printers is stable enough to be used as an intrinsic feature. However, line pattern is used to highlight the banding artifacts in their experimentations. Text-only documents require more sophisticated processing. Wu et al. [55] proposed a model of geometric distortion. The authors considered

four types of geometric distortions: translation, rotation, scaling and the distortion caused by the imperfection of the paper feeding mechanism. The last one is related to the phenomenon where ideal parallel lines appear as intersecting lines in the printed version. Considering that translation and rotation are inevitable during the print-and-scan process and do not characterize intrinsically the printers. Only parameters of the model, which relate to the two other distortions, are selected as features for SVM classification. Bulan et al [56] also exploited geometric distortions by considering the local variations of halftone dots within a cluster. The authors proposed to compared dot positions extracted from the printed image and estimated dot positions before printing. Some authors tried to study printed characters. In fact, printed characters from different printers may differ in toner density distribution, morphological characteristics and the roughness of edge profiles [57]. Studying the quality of printed characters can also provide useful information for forencics [58, 59, 60, 61, 62]. Approaches based on printed characters analysis, if the accuracy is high enough, can also help expose locally printed document forgeries.

2.4.5 Proactive approaches

In proactive approaches, watermarking information is added into the document before or during the printing process. The added information permits to verify after that authenticity or the integrity of printed documents. Watermaking techniques can be categorized into *robust watermarking* and *fragile watermarking*. Robust watermarks can resist to print-and-scan process and is often used for copyright protection. Meanwhile, fragile watermarks is sensitive to any operations, especially tampering. The latter is more suitable for content authentication. Most of color laser printers in the market add "invisible" tracking information into every page they print [63]. The tracking information is coded by patterns of yellow dots and they can encode a lot of information such as the printer's identity, the printing time,... Special inks or special papers can also be used, especially for copyright protection. Special characteristics of these inks or papers can be considered here as watemarking information. They are usually used for special documents such as certificates, banknotes,... Ibrahim et al. [64] proposed a robust watermarking technique for printed document. The technique uses a 512×512 gray scale image as a cover image, which is divided into watermark-size non-overlapping partitions. The embedded watermark is a 8×8 gray scale image. The cover image is known during both the embedding and extracting process of the watermark. For each partition of the cover image, the watermark will

be multiplied by an adaptive regulation factor and then added into the partition to form the watermarked image. The regulation factor is selected to maximize the peak signal to noise ratio between watermarked and cover images. The important drawback of the technique is that the cover image and the set of utilized regulation factors are required for the watermark extracting operations. The watermarked image is supposed to be placed somewhere within the document to protect its copyright. Varna et al. [65] proposed to hide watermarked data by simply adding/removing two groups of pixels into/from the left edge of every character which has a vertical stroke on their left (for example r,i,p,L...), Figure 2.9. The two groups of pixels are equidistant from the center of the related stroke. By adding/removing pixels into/from a character, one bit of embedded information can be encoded. The encoding capacity depends therefore on the number of documents' characters. The size and the shape of the group of pixels added or removed can be chosen to trade off robustness for imperceptibility. For the extracting operation of embedded data, all the embedded characters are first identified and cropped, the embedding operation applied on each character will be identified using a correlation-based detector. Many fragile watermarking

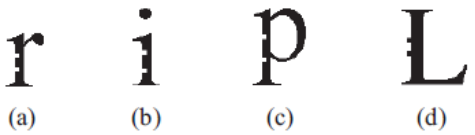


Figure 2.9: Ilustration of Varna el al.'s solution for data hiding in printed text documents.

solutions have been also proposed to protect printed documents. Picard et al. [66] proposed a solution based on creating digital images with specific properties, called a Copy-Detection Pattern (CDP) that is printed on arbitrary documents, packages. The pattern is sensible to the loss of information when documents go through a print-and-scan process. By measuring the amount of information contained in a scanned CDP, a detector can decide the authenticity of the document. Reliable performance measurements, based on a Neyman-Pearson hypothesis test, of the authentication system proposed by Picard are then given in a work of Ho et al. [67]. Baras et al. [68] proposed an application of this graphical code in protecting 2D matrix barcodes. Tkachenko et al.[69] proposed to substitute the black modules of standard QR code with a set of specific textured patterns in such a way that a private message can be encoded, which creates a second level of storage. The texture patterns are chosen to be sensitive to print-and-scan process. The public level of this code is read as normal as the one of the standard QR code. The private level is decoded by maximizing

the correlation values between each black module with the set of initial (numeric) patterns then linking it to the corresponding codeword. A copy attack implies two successive print-and-scan processes, which degrade the patterns. The authenticity of the code is decided by thresholding the mean of the mentioned correlation values. The limit of this approach is that the authentication process required an exchange of original numerical texture patterns. Wang et al. [70] exploited the particular optical characteristics of K, one of the four printed ink colors of a CMYK printer, the only color which can be rendered in infrared. The authors introduced an infrared watermarking technique to embed hidden information into the explicit graphic QR code, which permits its authentication. Teraura [71] also proposed to exploit the same characteristics of black ink under infrared to introduce a Double-Encode Two-Dimension code used for counterfeit detection purpose. Vongpradhip et al. [72] proposed to protect printed QR code by embedding an invisible watermark in the frequency domain. The authors in [73] and [74] suggest inserting a secret message by using the error correction capacity of the QR code. This manipulation decreases the error correction capacity of the standard code. In the section 4.2, we also proposed a fragile watermarking technique to protect printed QR codes from being cloned. The white background of QR codes is simply substituted by a particular random texture, which is sensitive to the print-and-scan process. A statistical detector is also proposed to authenticate the watermarked QR codes.

2.5 Overview on Digital Image Forgery Detection

With the emergence of technology, digital images have become an essential source of data in communication media. In the past, in order to manipulate an image generated by film cameras, professional knowledge and sophisticated dark-room equipments are required, which makes it difficult for average users. Nowadays, it becomes very easy to manipulate digital images with the availability of many of advanced image handling softwares. Digital images become more and more vulnerable against forgery. A digital image can be forged and served for malicious purposes, which can cause different adverse consequences for the society. For example, manipulated images can be used by malicious people to mislead a police investigation or to mislead the public on a social network. Therefore, it becomes very important to have solutions to verify the authenticity, the integrity and the truthness of digital images. Digital image forgery detection has become a research field which attracts a lot of

attention. This section introduces an overview of different types of digital image forgery and different existing solutions for image forgery detection.

2.5.1 Types of digital image forgery

There are many post-processing operations that can be used for digital image forgery. In practice, forging an image involves in combination many techniques. In this section, some common types of image forgery technique are introduced.

Copy-move

Copy-move forgery is the most popular and common type of digital image forgeries. It involves copying of some region in an image and then moving and pasting it into some other regions within the same image. Figure 2.10 shows a typical example of copy-move forgery. The small aircraft was duplicated in the forged image. For some malicious reasons, this operation tries to falsify the quantitative information transmitted by the image. Copy-move can also be employed to achieve a content-aware filling operation, see Figure 2.11 for example. This operation is often utilized for suppressing unwanted objects or part of it in the image.

(Original) **(Forged)**

Figure 2.10: Example of copy-move forgery.

The forged region can also be scaled by considering the perspective effects, which makes it look more natural under human eyes. Because the copied parts come from the same image, if there is no geometric transformation, their essential properties such as noise, color, dynamic range, texture will be compatible with the rest of the image, which makes the recognition process based on these properties troublesome.

(Original) **(Forged)**

Figure 2.11: Example of content-aware filling.

Image splicing

Image splicing is another common type of digital image forgeries. It consists of cutting a region of one image and then pasting it into another image or into another region of the same image to create a forged image. When splicing is performed precisely, the borders between the spliced region can visually imperceptible. Figure 2.12 shows an example of splicing forgery. The deer in the first image was cut out and pasted directly on the second image to create the forged one.

(Original 1) **(Original 2)** **(Forged)**

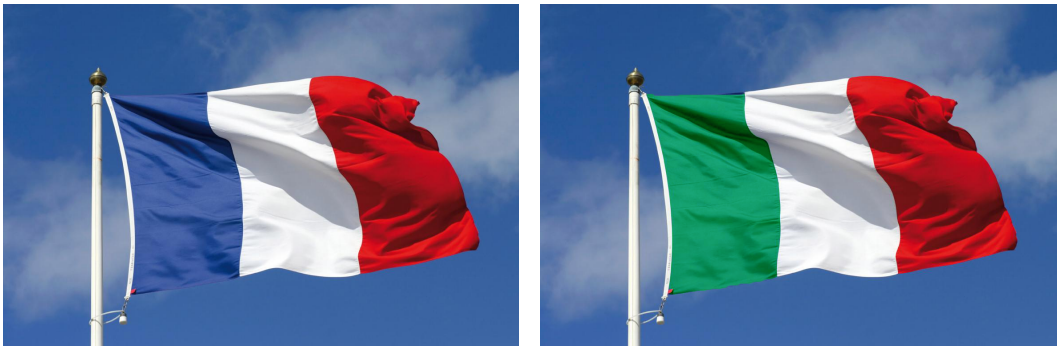
Figure 2.12: Example of splicing forgery.

Image resampling

To obtain a convincing forged image, forged regions sometimes have to undergo geometric transformations, such as rotation, scaling, stretching, skewing, or flipping, etc. These geometric transformations involve certain interpolation operations which always lead to the resampling of pixels within the forged region. That's why we used the term *image resampling* to refer all forgeries which are related to geometric transformations.

Local adjusting

Local adjusting consists of applying locally an adjustment in an image to create a forged one. When applying an adjustment in a region of image, different properties such as color, intensity or texture of the adjusted region can be modified. Figure 2.13 shows an example of forged image created by using local adjustment of color. The original image, on the left, presents a French flag. By simply adjusting the color the blue part of the French flag, we obtained an image of an Italian flag. The forged image transmit information totally different from the original one.



(Original) **(Forged)**

Figure 2.13: Original image and forged image created by using local adjustment of color.

2.5.2 Digital image forgery detection solutions

Figure 2.14 gives a categorization schema for existing digital image forgery detection solutions. Different solutions can be regrouped into two main categories: **active approaches** and **passive approaches**. Examples of active approaches are digital signature and watermarking. In these approaches, certain digital information is embedded into images and then is served as means to authenticate the content of these images. The embedding process occurs systematically during the creation of digital images. The application fields of these approaches are very limited. Active approaches are intrusive. Meanwhile, passive approaches are non-intrusive. These approaches do not need any extra embedded information, such as watermark or signature. These approaches exploit only information that is inherent to images. In this work, we have only focused on passive approaches. We can divide passive approaches into two groups: **forgery based** and **regularity base** approaches. Forgery

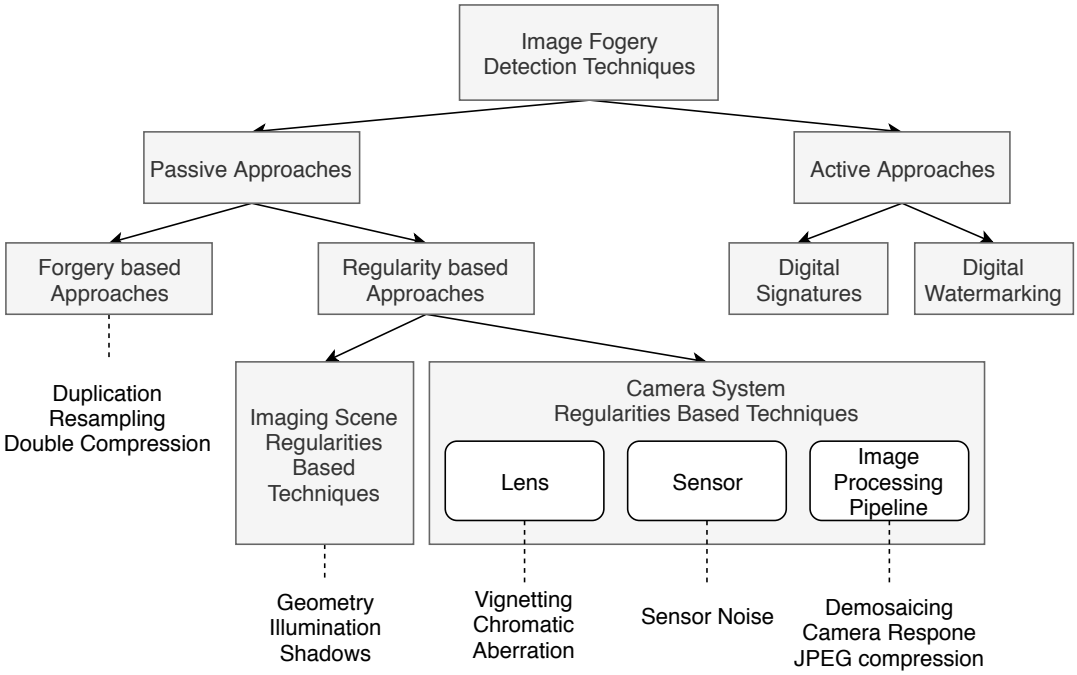


Figure 2.14: Categorization of image forgery detection techniques.

based approaches focus on some particular characteristics which are probably carried out by a given type of forgery on the image, such as redundancy in a copy-move forgery or correlation in a resampling operation.. Regularity based approaches are based on the assumption that images possess certain regularities or invariants, which can be disturbed by forgering. The difference in the measurements of the interested regularities or invariants permits to detect and localize the forged region. Based on the source of exploited regularities, these approaches can be categorized into two groups: approaches based on regularities of the imaging scene and approaches based on regularities of the camera system. Most of the existing image forgery detection solutions target some specific types of forgery since that a single operation may disturb only some specific types of regularities. There is no single solution which can detect all types of tampering. In practice, different solutions are often used together to detect a wider range of forgery.

Forgery based approaches

Some types of forgery introduce typically in images some particular characteristics or artifacts, which can be exploited for image forgery detection. Forgery based approaches rely on properties of the image itself and do not impose any assumption about the imaging scene and the camera system. Copy-move systematically introduces content redundancy in forged images: the presence of duplication objects

(Figure 2.10) or the presence of high correlated background regions (Figure 2.11). In the Figure 2.11, the forged region is disguished within its very similar qualitatively surrounding unforged regions, which makes it look naturally under human eyes and then hard to be perceived. Intuitively, we can simply divide an image into fixed-size overlapping blocks and search for numerically identical pairs of blocks to detect copy-move forgery. This naive solution was referred as exact matching in [75]. However, in practice, the forged region was often intentionally retouched by the forger or systematically degraded by JPEG compression, which makes it a little bit different from the original one. This solution is therefore not efficient. A comparison pixel-by-pixel between blocks also requires prohibitively expensive computational resources. Therefore, to obtain more efficient and less expensive copy-move detection solutions, a common strategy has been adopted by several existing works. First, the studied image is divided into small fixed-size overlapping blocks. A specific algorithm is applied to obtain a reduced dimension representation of each individual block. This representation provides a convenient space which can be used for block comparison. Some detrimental information is removed from the reduced dimension representation of the blocks, which makes the block comparison robuster against corrupting noise or minor degradations. The elements of the new representation of each block will then be rearranged to form a vector, referred as detection features. The matching of similar blocks is then simply realized by looking for identical rows from the lexicography ordered matrix which contains the extracted features of every block. Fridrich et al. [75] have employed quantized DCT coefficients as the detection features. 16×16-size blocks have been used in their algorithm. The small block size generated many false matches. To obtain a better copy-move detection, the author has considered the mutual positions the matching blocks. Some short of shift-vector has been used by the authors to account for the translation movement between matching blocks. By assuming that there was no geometric transformation applied to the forged regions, the matching between forged region and its original counterpart can be represented by a set of identical shift-vectors. Popescu et al. [76] have simply applied the principal component analysis (PCA) transformation to obtain the reduced dimension representation. Kang et al. [77, 78] have proposed to use Singular Value Decomposition (SVD) to extract the detection features. Singular value features can represent algebraic and geometric invariant properties of an image, which make them very useful in pattern recognition and image analysis. These features are robust against different noises and affine transforms. Lin et al. [79] have proposed to represent each 16×16-size blocks by a 9-dimensional vector. Each block is divided into 4

non-overlapped equal size sub-blocks. Average intensity of the big block, f_1, and of each sub-block, and the difference between the average intensity of each sub-block to f_1 constitute the 9 elements of the detection features. Zhang et al. [80] have proposed to transform images with Discrete Wavelet Transform (DWT) and employed only the lowest frequency subband for copy-move detection. The selected subband image is then divided into four non-ovelapping subimages. Phase correlation is used to compute the spatial offset between forged region and the original one. The copy-move regions are located by the idea of pixel-matching, which is shifting the input image according to the spatial offset and calculating the difference between the image and its shifted version. Furthermore, we can mention several other copy-move detection solutions, such as: [81] using SIFT features, [82] using SURF features, [83] basing on DWT and DCT, and so on. This section does not try to propose a complete state-of-the-art of copy-move detection solutions. In the case of splicing or copy-move, the forged regions are often scaled or rotated to be proportional and coherent with the neighboring unforged regions. These geometric transformations lead to a local resampling of the image grid. This resampling creates inconsistencies between regions in the forged image. Therefore, the presence of local resampling can be considered as evidences of image manipulations. Basing on resampling detection, several solutions have been proposed. Popescu and Farid [84] have exploited specific statistical correlations caused by resampling and described how these correlations can be automatically detected in any portion of the image. Mahdian and Saic [85] proposed a solution to detect traces of resampling using periodic properties brought into the image and its derivatives by low-oder interpolation polynomials. The author has also employed the Radon transformation to detect traces of affine transformation. The periodic artifacts caused by resampling have been also exploited in [86]. The boundaries of forged regions contain pixels from both forged regions and original ones. Brutal change of pixel values may present within the boundaries if no additional retouching operation is applied on. However, if the forger try to smooth this change, other artifacts may be generated. Without describing the chararistics of these boundaries, Zhang et al. [87] have proposed a Shallow Convolutional Neural Network capable of distinguishing these from original edges in images. The neural network was claimed to well perform on low resolution images.

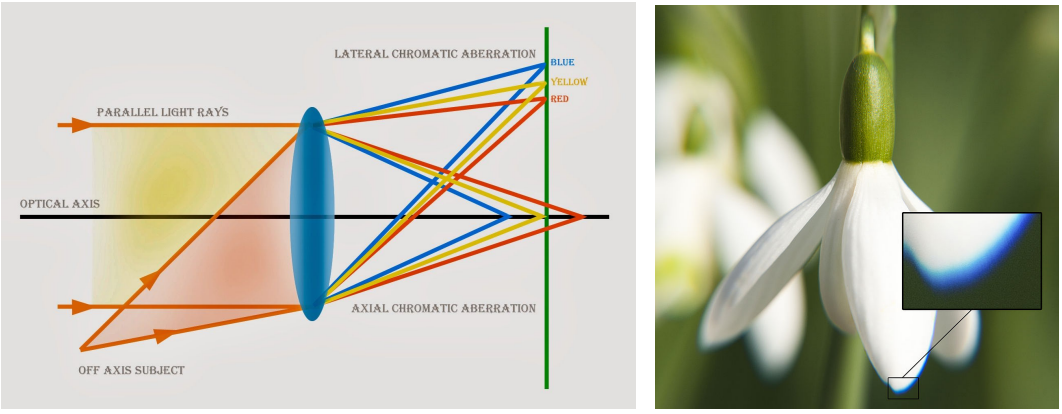
Approaches based on regularities of the imaging scene

Incident light from the imaging scene is captured by the camera system to form an image. Since rays of light obey certain physical laws, the first source of image regularities is the imaging scene itself. These regularities can be interpreted by the geometric relationship or the consistency in the lighting conditions between different objects in the imaging scene. The imaging process will map the 3D imaging scene to 2D points in an image. This mapping can be modeled using projective geometry. The relationship between 2D points in an image has to satisfy the projective geometry model. Forged regions within an image can come from another scene which was generally captured in the image under another perspective. Geometric relationship between pixels in forged region is therefore different from the rest of the image. This relationship can be studied through different properties such as lengths, angles, straight lines, point clusters, etc. Connoter el al.[88] proposed a solution for detecting photo manipulations on signs and billboards by analyzing text under perspective projections. The solution starts by identifying stable features of texts in the image using SIFT descriptors. The SIFT features are then used to estimate the homography between the text and a fronto-parallel version of the same text in the same font. After estimating the homography, the text in the image is warped according to the homography and an image-based error metric is used to estimate authenticity. Considering that the lighting conditions are constant over an imaging scene, all objects in the scene have to be illuminated by the same light sources. Therefore, the lighting conditions' information of different objects in an image should be coherent. When a forged image is created, it is often difficult to ensure this coherence. Farid et al. [89, 90, 91] proposed different models for representing lighting environment which are then utilized for detecting forgery in images. Another aspect of the imaging scene, the shadows, which are directly related to the lighting conditions and 3D scene structure, can also provide useful information for forensics. Zhang et al.[92] proposed a image forgery detection solution based on analyzing the shadow consistency in images.

Approaches based on regularities of the camera system

When capturing a scene into an image, different parts of the camera system, such as a lens, sensors or the image processing pipeline, can leave in their final output some relevant features, which are consistent overall the captured image. Any manipulation, especially forgery or tampering operations, can introduce inconsistencies

between different regions of the manipulated image. The camera lens system is a very important part of a camera. Its function is guiding light rays from the imaging scene to the surface of the camera sensor. In an ideal camera system, light rays from a point in the scene that pass through the lens system have to converge in a same point on the sensor. However, because lenses have different refractive indices for different wavelengths of light, real optical systems fail to perfectly focus light of all wavelengths to the same convergence point. This optical failure creates the so called chromatic aberration effect in image. There are two types of chromatic aberration: axial and



(a) **(b)**

Figure 2.15: (a) Illustration of the chromatic aberration, and (b) lateral chromatic aberration effect in image.

lateral, Figure 2.15a. Axial aberration occurs when different wavelengths are focused at different distances from the lens. The axial chromatic aberration leads to colored areas in the images that arise because not all three colors can be displayed in good focus. Lateral aberration occurs when different wavelengths are focused on different positions in the focal plane. Lateral aberration can be seen as a blur and a rainbow edge in areas of high contrast, Figure 2.15b. Lateral chromatic aberration (LCA) can be modeled as an expansion/contraction pattern of the color channels about the optical center. Farid and Johnson [93] first introduced the idea that LCA can be used to detect image forgeries. The authors proposed an algorithm to find a global estimate of the expansion/contraction pattern over the whole image by considering different pairs of color channels. Local estimation of the pattern is then performed over non-overlapped blocks in the image with the same algorithm. Forgeries are detected by comparing local observations of expansion/contraction pattern to the global model then identifying localized inconsistencies. Mayer and Stamm [94] have recently proposed a statistical model which can capture the inconsistency between global and local estimates of LCA. The statistical model is then used to pose forgery detection

as a hypothesis testing problem. Yerushalmy et al [95] also exploited image artifacts caused by the chromatic aberration as indicators for evaluating image authenticity. The authors claimed that not only the lens but the sensor can introduce chromatic aberration in images. Optical vignetting, which corresponds to a reduction of image brightness or saturation toward the periphery compared to the image center, is also a type of lens distortion. Optical vignetting is caused by the physical dimensions of the lens system. Considering that patterns of vignetting are characteristics of lens models, Lyu [96] proposed a method exploiting these patterns for digital image authentication. Camera sensors consist of an array of photo detectors, which absord incident photons and convert into electrons. The response of these detectors are not uniform due to the imperfection of their manufacturing process. This variation, referred as the photo response nonuniformity noise (PRNU), can be considered as fingerprints of the camera sensor. The PRNU noise can be modeled and used for camera identification [97], and image forgery detection [98]. Thai et al. [99, 100] have proposed different heteroscedastic noise models for RAW, TIFF and JPEG images, which were employed for camera model identification. Their works are potentially exploitable for image forgery detection. To obtain a properly exposed image, the camera sensor must be held stably during the exposure time. Any motion of the camera or of the imaging objects during this time will create blur in the image. Blur caused by the movement of camera will affect every pixel consistently in the images. Kakar et al. [101] proposed to use motion blur estimation through image gradients in order to detect inconsistencies between different regions in forged images. Different operations of the camera's image processing pipeline can also introduce in image several invariants which have been exploited in many works for digital image forgery dectection. Popescus et al. [102] proposed to quantify the statistical correlations introduced by CFA interpolation (a.k.a demosaicing), and describe how these correlations can be used to expose image forgeries. The authors employed the expectation-maximization algorithm to fit a linear model of nearby pixels correlations. CFA interpolation will introduce periodic correlation patterns over an image. Forgered region in an image can be figured out by abrupt changes of the correlation patterns. The traces of democaising have been also exploited in some other works [103, 104] for photorealistic computer generated images detection and for image forgery detection. Different operations, such as gamma correction, white balancing, will introduce a non-linear relation between the amount of incident light and the image pixel values. This non-linearity est characterized by the Camera Response Function (CRF). The CRF should be consistent over a natural image. The inconsistency of the CRF between different

regions of the image can also be considered as traces of image forgery [105, 106]. For the ease of storage and transmission, images are often compressed using the JPEG compression algorithm. This lossy compression operation introduces several artifacts in images. Agarwal and Farid [107] described a JPEG artifact, termed as JPEG dimple, which arises differently depending on the choice of the mathematical operator (rounding,flooring or ceiling) used to convert DCT coefficients from floating-point to integer values. The author showed that JPEG dimples can be used to reliably detect a wide range of image manipulation. The blocking artifacts, caused by the blocking processing during JPEG compression, are usually mismatched when the forged region is inserted. Inconsistencies in the blocking artifacts have been studied and exploited in different works [108, 109, 110] for image forgery detection. The JPEG compression results images in which 8×8 non-overlapped blocks are quantized in the DCT domain. The quantization and dequantization operations create some periodic properties in the value of the dequantized DCT coefficients. This periodic properties of blocks within the forged regions are often lost or inconsistent with the ones of original blocks. This inconsistency is a very useful evidence of image forgery. The workflow for creation forged images often finishes by resaving forged images under JPEG format. This operation involves a JPEG compression. If the original image is in JPEG format, a double compression is involved. The inconsistency mentioned previously will be attenuated after the second compression. However, it can always be sophistically exploited for forgery detection. Many existing solutions can be mentioned here [111, 112, 113, 114]. In the chapter 5, a new and simple image forgery detection solution is proposed. The proposed solution exploits particular properties left by JPEG compression and double compression on the quantized DCT coefficients.

2.6 Conclusion

By the advent of technologies, a myriad of applications using digital images have been introduced. Systems exploiting digital images for specific applications, particularly authentication systems, have to face up different security challenges. The chapter 3, we introduce a novel face presentation attack detection solution based on a statistical model of image noise. In the chapter 4, by inspiring the previous face PAD solution, we propose a watemarking technique to secure printed QR codes, which would be used for authenticating commercial products. The watemarked QR codes are sensitive to the print and scan process. Any attempt to create a falsified QR code

based on an existing one can be detected. For both of the two applications, the detection problems are casted in the framework of hypothesis testing. Different statistical models in both spatial and frequency domains have been proposed.

Digital images can be easily manipulated by powerful computer softwares. In different contexts, such as during police investigation, manipulated images which transmit usually erroneous information can introduce different adverse consequences. In these cases, we have to certify the authencity and the integrity of digital images before exploiting them. In the chapter 5, we introduce an efficient solution for detecting falsifications in digital images based on the particular characteristics of JPEG compression in the DCT domain.

Chapter 3

Face PAD solution based on a statistical model of image noise

Contents

3.1 Introduction

There are many solutions to approach the face PAD problems as we described in the section 2.3. In this work, we proposed a face PAD solution, which targeted facial authentication systems using digital still cameras. The proposed solution exploits noise extracted from face images. Perceptually, image noises are undesired variation in

pixel intensity value, which non-uniformly corrupts the smooth surface in an image and degrades the image quality. Image noises can come from various sources during the acquisition process: lighting condition, imperfect instruments, photon conversion, compression, and transmission losses, etc. In this chapter, noise from these sources is referred as **acquisition noise**. Micro-textural information presents on the surface of objects captured in an image can also be interpreted as noise in the final output of the image acquisition process. The nature of these textures is various. For example, they can be small bristles in a human face, imperfections of printed photo due to printing quality or the moiré pattern presenting on the surface of an LCD screen. We referred this micro-textural information as **textural noise**. Acquisition noise values depend mainly on the acquisition process. They vary according to the choice of acquisition device, the lighting conditions, and the camera settings (ISO's configuration, JPEG compression quality, output's resolution...). Meanwhile, the nature of textural noise significantly relies on the scene captured in images. It varies differently according to the nature of the scene, from a bona-fide face to a presentation attack one. So that, textural nose can provide us interesting information to classify image scenes. In this precise problem, we can use textural noise to distinguish images real face and the ones captured during a face presentation attack. Textural noise is indistinguishable from acquisition noise. In fact, noise that we can estimate from images is eventually a fusion of the acquisition noises and the textural one. It is hard or even impossible to separate the two noises. So that, in order to investigate the textural noise, we proposed to consider the estimated noise while the acquisition noise is maintained statistically stable between the images. The statistical stability of acquisition noises can be expectedly assured by holding unchanged acquisition conditions.

3.2 Modeling

3.2.1 Noise Local Variance (NLV) model

Denote $z_i, i \in \mathcal{I}$ the noise value estimated for the i-th pixel of the concerned image, here $\mathcal{I}$ is the set of indexes of all pixels. We assume that $z_i, i \in \mathcal{I}$ are independent. Following the heteroscedastic noise models proposed by Thai et al. [99, 115], we know that the distribution of acquisition noise can be approximated by a zero-mean Gaussian one, where the standard deviation depends on pixel expectation. We generalized this model by considering the impact of noise caused by textural information,

which is assumed to be a zero-mean normal random variable with the standard deviation characterized by σ_s. Then, the standard deviation of z_i is given by $\sqrt{f_\sigma(\mu_i)}$, where μ_i is the pixel expectation and $f_\sigma(\mu)$ is defined as follows. For a RAW image:

$$f_\sigma(\mu) = a\mu + b + \sigma_s^2 \tag{3.1}$$

where (a, b) are constant, considered as a fingerprint of the camera. For a TIFF image or JPEG image with high quality factor $(QF \geq 70)$:

$$f_\sigma(\mu) = \frac{1}{\gamma^2}\mu^{2-2\gamma}(\tilde{a}\mu^\gamma + \tilde{b} + \sigma_s^2) + \frac{\Delta^2}{12} \tag{3.2}$$

where γ is the gamma correction factor, $(\tilde{a}, \tilde{b})$ is the camera fingerprint. $(\tilde{a}, \tilde{b})$ differs from the above (a, b) due to demosaicing, white balancing operations, Δ is the quantification step which characterizes the eventual quantification noise. Given image is divided into 8x8-pixel blocks. Denote S^2 the sample variance of noise calculated within a block; it is defined as follows:

$$S^2 = \frac{1}{63}\sum_{i=1}^{64}(z_i - \bar{z})^2 \tag{3.3}$$

where i is actually the index of pixel within the block, $\bar{z}$ represents the average of noise values within the block, defined by:

$$\bar{z} = \frac{1}{64}\sum_{i=1}^{64} z_i. \tag{3.4}$$

We obtain the following result, demonstrated in the Appendix A.1.

Theorem 1 (Distribution of S^2) *Within a block, when μ_i is sufficiently close to each other, we can approximate the distribution of S^2 by a Gamma distribution $\mathcal{G}(\alpha, \beta)$, where α is its shape factor, β is its scale factor, (α, β) is signal-dependent and:*

$$\alpha\beta \approx \frac{1}{64}\sum_{i=1}^{64} f_\sigma(\mu_i). \tag{3.5}$$

3.2.2 NLV Model & The Face PAD Problem

A facial image is rich in content, and so the behavior of noise over the whole image is quite heterogeneous. To deal with face presentation attacks by exploiting the

proposed NLV model, we have to limit our study in some relevant regions of the image. Textural information intervenes directly on the image content. In a facial image, the skin does not have the same texture as the hair, the clothes or obviously the background. Additionally, it can be noted from the theorem 1, the distribution of S^2 is signal-dependent. The difference in textural information causes the difference in content and then in the behavior of the proposed NLV model. Therefore, we have to focus only on some region which contains specific textural information, which has a significant size and which is persistent overall studied images. For facial images, the interested region should be the skin one. In this work, the extraction of the skin region in the image can be realized based on skin color likelihood method proposed in [116].

The proposed NLV model is signal-dependent. Image content should be studied to extract correct data for our model. The image content can be obtained by subtracting noise from the natural image. Due to the non-reversibility of noise production, it's impossible to know precisely image noise and so image content. However, we can estimate image noise thanks to a denoising process. In this work, we employed the wavelet-based denoising technique described in the Appendix A.3. Figure 3.1 resumes different operations we have to do with a facial image to extract valuable data for our proposed problem. First, from the full-color image, locate the skin zone within image, a skin mask is obtained. In this work, it is proposed to analyze the green color channel for the next operations. Another color channels can also be employed alternatively, but it is better to use the green one because it contains more information due to CFA demosaicing process. Then, the one channel image is going through the denoising process to obtain *image noise* and *image content*. Image content and image noise are then segmented into 8x8 non overlapping blocks. Each block in image noise has its counterpart at the same location on the image content. Denote $\bar{\mu}_{i_b}$ and v_{i_b} respectively the empirical mean and variance of content values within the i_b-th block, $i_b \in \mathcal{I}_b$ and $\mathcal{I}_b$ is the set of block index. For the sake of simplicity, by omitting the block index, denote $\tilde{\mu}_i, i \in \{0, 1, 2, .., 63\}$ is the estimated content pixel values, we have:

$$\bar{\mu}_{i_b} = \frac{1}{64} \sum_{i=0}^{63} \tilde{\mu}_i \tag{3.6}$$

$$v_{i_b} = \frac{1}{64} \sum_{i=0}^{63} \left(\tilde{\mu}_i - \bar{\mu}_{i_b}\right)^2 \tag{3.7}$$

We have to eliminate the blocks which do not satisfy the necessary condition of the

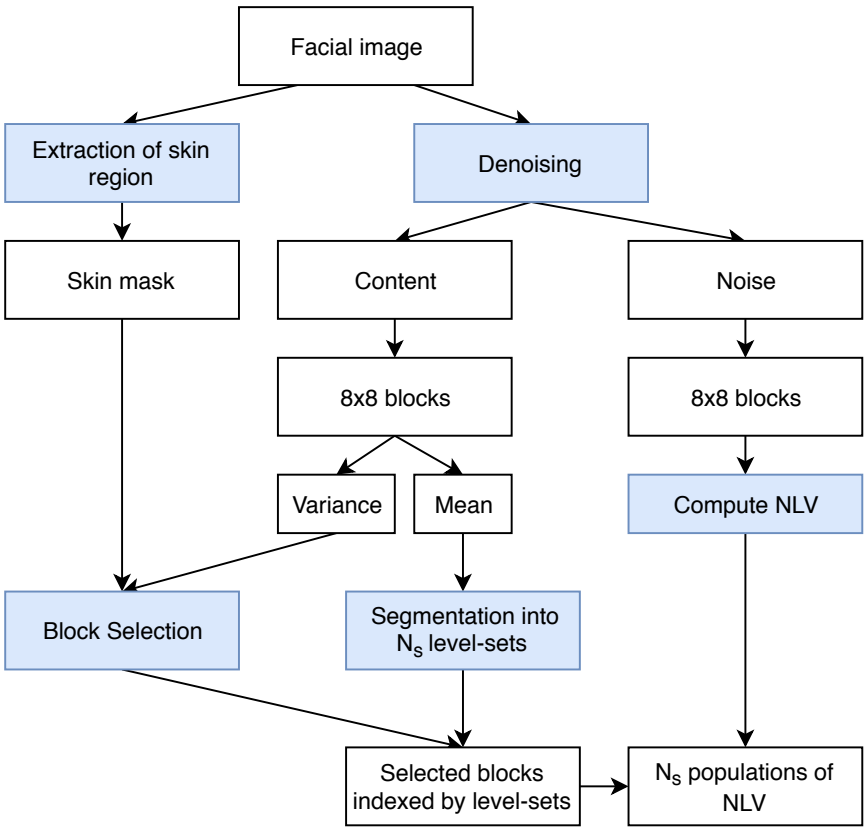


Figure 3.1: Data extraction flowchart.

theorem 1. All the blocks whose content pixel values deviate significantly should be eliminated. We used $\sqrt{v_{i_b}}$ as an estimate of this deviation. The i_b-th block will be eliminated if v_{i_b} exceeds some threshold τ_b, which can be defined analytically based on the Linderberg's condition [117]. For the sake of simplicity, in this paper, we just selected a small enough value for τ_b. We named the blocks which stay after elimination as homogeneous blocks. Homogeneous blocks situated within the skin region are interested in the proposed solution. For every interested block, the estimation of noise local variance S^2 is computed. It is noted that these blocks do not have the same content because of the variation of pixel intensity. The distribution of S^2 cannot be compared between these blocks. It is proposed to regroup these blocks into subsets, referred as *level-set*, basing on their empirical content mean $\bar{\mu}_{i_b}$ so that blocks within a level-set have comparable content. The whole range of image intensity is divided into N_s non-overlapping equal intervals. The width of each interval, denoted by Δ_s, is derived directly from the value of N_s. In fact, for an image which has pixel intensity varying from 0 for absolute black to 1 for absolute white, we have:

$$\Delta_s = \frac{1}{N_s} \tag{3.8}$$

We attribute all the interested blocks into N_s level-sets. Block i_b is in the k-th level-set if its empirical content mean, $\bar{\mu}_{i_b}$, drops in the k-th interval defined previously. Denote $\mathcal{I}_l$ the set of index of all blocks belonging to the l-th level-set. Without loss of generality, denote $S_j^2, j \in \mathcal{I}_l$ the NLV of the j-th block. Flowing theorem 1, S_j^2 is considered as a sample of a Gamma random variable characterized by (α_j, β_j), which depends on the content of j-th block, and we have that:

$$\alpha_j \beta_j \approx \frac{1}{64} \sum_{i=1}^{64} f_\sigma(\mu_{ji}) \tag{3.9}$$

where the i-index in μ_{ji} is related to the index of pixels within the block. It is noted that $S_j^2, j \in \mathcal{I}_l$ are not identically distributed because of the difference of blocks' content. However, when Δ_s is small enough, the difference between blocks' content can be ignored, and we can assume that $S_j^2, j \in \mathcal{I}_l$ are identically distributed. They form a population of a random variable following approximately a Gamma distribution. Denote (α_l, β_l) the parameters of the latter distribution, l refers to the index of level-set, (α_l, β_l) represents all the set of parameters $(\alpha_j, \beta_j), j \in \mathcal{I}_l$ in the equation 3.9. For RAW images, we have:

$$\alpha_j \beta_j \approx a \frac{1}{64} \sum_{i=1}^{64} \mu_{ji} + b + \sigma_s \tag{3.10}$$

For a specific texture and segmentation sufficiently narrow, we can assume that σ_s and $\frac{1}{64} \sum_{i=1}^{64} \mu_{ji}$ are approximately constant. Then, $\alpha_l \beta_l$ is constant. In other words, there is a linear relation between α_l and β_l^{-1} and this relation is maintained for all images of a typical texture (i.e., real face). For two different textures (i.e., real face and falsified face images), this relation may differ due to the probable difference in σ_s and due to the change in the repartition of image content. For TIFF or JPEG images, $\alpha_j \beta_j$ can also be represented in function of μ_{ji} and σ_s. For a specific texture, we can also deduct approximately a linear relation between α_l and β_l^{-1}, which is maintained over all images of a typical texture.

3.3 Proposed face PAD solution

3.3.1 Hypothesis Testing Formulation

In practice, due to the specific color range of skin, there will be level-sets that contain a tiny number of interesting blocks, especially the low or high-range level-sets. So, among the above N_s level-sets, we opt to study the $M < N_s$ level-sets in the middle range, which have the most significant number of samples. Let's X_l, $l \in \{1, 2, ..M\}$ the populations of noise block variance of the M selected level-sets. Let's n_l the size of the population l. We have demonstrated in the section 3.2 that X_l follow a Gamma distribution and, for images of a given type of textural noise, there is a linear relation between the shape parameter and the inverse of scale one of this distribution. Then, for any l and a given type of textural noise, we have that:

$$X_l \sim \mathcal{G}(\alpha_l, \beta_l) \tag{3.11}$$

where $\alpha_l = a_{l0}/\beta_l + b_{l0}$, and (a_{l0}, b_{l0}) are constant.In matrix form, we have:

$$\begin{bmatrix} 1 & -b_{10} & 0 & ... & ... \\ ... & ... & ... & ... & ... \\ ... & ... & 0 & 1 & -b_{M0} \end{bmatrix} \begin{bmatrix} \alpha_1\beta_1 \\ \beta_1 \\ ... \\ \alpha_M\beta_M \\ \beta_M \end{bmatrix} = \begin{bmatrix} a_{10} \\ ... \\ a_{M0} \end{bmatrix} \tag{3.12}$$

Denote respectively B, θ and A the three terms from left to right of the equation (3.12). The Mx2M matrix B and the M-length vector A are supposed to be known. Their estimation is proposed in 3.3.2. The 2M-length vector θ is unknown. We aim to detect falsified image based on the proposed model. Our detection problem is cast in the framework of hypothesis testing. For a face bona-fide image, the model is supposed to satisfy the equation (3.12). For falsified image, due to the difference in textural noise, the equation (3.12) would not be verified. Consequently, the goal of the test is to decide between two hypothesis defined as follows:

$$\begin{cases} \mathcal{H}_0 : B\theta = A \\ \mathcal{H}_1 : otherwise \end{cases} \tag{3.13}$$

In this paper, we focus on guaranteeing a prescribed false alarm probability. Let

$$\mathcal{K}_{\alpha_0} = \{\delta : \mathbb{P}_{\mathcal{H}_0}[\delta(X_1, X_2, ..X_M) = \mathcal{H}_1] \leq \alpha_0\} \tag{3.14}$$

be the class of tests with a false alarm probability upper-bounded by α_0. Here $\mathbb{P}_{\mathcal{H}_j[E]}$ stands for the probability of event E under hypothesis $\mathcal{H}_j$, $j \in \{0,1\}$. Our proposed statistical test of class $\mathcal{K}_{\alpha_0}$ is given in the section 3.3.3.

3.3.2 Estimation of linear parameters

The linear relationship between parameters of Gamma distribution, characterized by the matrix B and the vector A, are estimated from a set of training images, which are all bona-fide images. Each image of the training set is divided into level-sets. For each level-set, we estimate the parameters of the gamma distribution which fits the population of the set thanks to the Maximizing Likelihood Method. For a good estimation of the parameters, only the sets which contain more than than a predefined number of elements are considered. For each level-set, we obtain a set of estimated Gamma distribution's parameters. The linear relation between these parameters is then simply obtained by linear regression using the Mean Square Error method.

3.3.3 Proposed test statistics

Denote the j-th cumulant of the l-th population by κ_{lj}, see Appendix A.4 for a brief description of cumulant. We have that

$$\kappa_{l1} = \alpha_l \beta_l \tag{3.15}$$

$$\kappa_{lj} = (j-1)!\alpha_l \beta_l^j \tag{3.16}$$

$$\kappa_{l,j+1}/\kappa_{lj} = j\beta_l \tag{3.17}$$

Let denote the following measures

$$\eta_{l1} = \kappa_{l1}, \eta_{lj} = \kappa_{lj}/\kappa_{l,j-1}, \qquad j > 1 \tag{3.18}$$

$$\eta_l^T = (\eta_{l1}, \eta_{l2}, \eta_{l3}, \eta_{l4}) \quad l = 1, 2, ..M \tag{3.19}$$

$$\eta^T = [\eta_1^T, \eta_2^T, .., \eta_M^T] \tag{3.20}$$

where $.^T$ denotes the matrix transposition operation. We can easily see that

$$\eta_l = \omega \begin{bmatrix} \alpha_l \beta_l \\ \beta_l \end{bmatrix}, \quad l = 1, 2.., M \tag{3.21}$$

where

$$\omega^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix} \tag{3.22}$$

If we let $w = diag(\omega, \omega, .., \omega)$, obtained by concatenating diagonally M times the matrix ω , we can get the following linear relationship

$$\eta = w\theta. \tag{3.23}$$

Denote Σ the asymptotic covariance matrix of η. It can be defined by

$$\Sigma = J_2 J_1 V J_1^T J_2^T \tag{3.24}$$

with V, J_1 and J_2 given below:

$$V = diag(V_1, V_2, ..., V_M),$$

where

$$V_l = \frac{1}{n_l} \begin{bmatrix} \mu_{l2} - \mu_{l1}^2 & \mu_{l3} - \mu_{l2}\mu_{l1} & \mu_{l4} - \mu_{l3}\mu_{l1} & \mu_{l5} - \mu_{l4} \\ \mu_{l3} - \mu_{l2}\mu_{l1} & \mu_{l4} - \mu_{l2}^2 & \mu_{l5} - \mu_{l3}\mu_{l2} & \mu_{l6} - \mu_{l4}\mu_{l2} \\ \mu_{l4} - \mu_{l3}\mu_{l1} & \mu_{l5} - \mu_{l3}\mu_{l2} & \mu_{l6} - \mu_{l3}^2 & \mu_{l7} - \mu_{l4}\mu_{l3} \\ \mu_{l5} - \mu_{l4}\mu_{l1} & \mu_{l6} - \mu_{l4}\mu_{l2} & \mu_{l7} - \mu_{l4}\mu_{l3} & \mu_{l8} - \mu_{l4}^2 \end{bmatrix} \tag{3.25}$$

and μ_{lj} is the j-th raw moment of the l-th population. J_1 and J_2 are the Jacobians

$$J_1 = diag(J_{11}, J_{12}, ..., J_{1M})$$

$$J_2 = diag(J_{21}, J_{22}, ..., J_{2M})$$

where J_{1i} and J_{2i} correspond to the following transformations:

$$J_{1l} : (\mu_{l1}, \mu_{l3}, \mu_{l2}, \mu_{l4}) \longrightarrow (\kappa_{l1}, \kappa_{l2}, \kappa_{l3}, \kappa_{l4})$$

$$J_{2l} : (\kappa_{l1}, \kappa_{l2}, \kappa_{l3}, \kappa_{l4}) \longrightarrow (\eta_{l1}, \eta_{l2}, \eta_{l3}, \eta_{l4})$$

The elements of J_{1l} and J_{2l} are:

$$J_{1i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2\mu_{l1} & 1 & 0 & 0 \\ -3\mu_{l2} + 6\mu_{l1}^2 & -3\mu_{l1} & 1 & 0 \\ -4\mu_{l3} + 24\mu_{l2}\mu_{l1} - 24\mu_{l1}^3 & -6\mu_{l2} + 12\mu_{l1}^2 & -4\mu_{l1} & 1 \end{bmatrix} \tag{3.26}$$

$$J_{2i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\kappa_{l2}/\kappa_{l1}^2 & 1/\kappa_{l1} & 0 & 0 \\ 0 & -\kappa_{l3}/\kappa_{l2}^2 & /\kappa_{l2} & 0 \\ 0 & 0 & -\kappa_{l4}/\kappa_{l3}^2 & 1/\kappa_{l3} \end{bmatrix} \tag{3.27}$$

Let h be the sample counterpart of η and $\hat{\Sigma}$ be a consistent estimate of Σ. The best estimation of θ should minimize the following quadratic form:

$$Q = (h - w\theta)^T \hat{\Sigma}^{-1} (h - w\theta) \tag{3.28}$$

The gradient of Q is obtained below:

$$grad(Q) = 2w^T \hat{\Sigma}^{-1} w\theta - 2w^T \hat{\Sigma}^{-1} h \tag{3.29}$$

Under no restriction, $\hat{\theta}$ which minimizes the value of Q is therefore given by

$$\hat{\theta} = (w^T \hat{\Sigma}^{-1} w)^{-1} w^T \hat{\Sigma}^{-1} h \tag{3.30}$$

Under $\mathcal{H}_0$, where θ should satisfy additionally the condition (3.12), the value $\tilde{\theta}$ that minimizes Q is obtained as below, see Appendix A.5 for demonstration:

$$\tilde{\theta} = \hat{\theta} - (w^T \hat{\Sigma}^{-1} w)^{-1} B^T (B(w^T \hat{\Sigma}^{-1} w)^{-1} B^T)^{-1} (B\hat{\theta} - A) \tag{3.31}$$

Denote Q_0 and Q_1 respectively the minimum values of Q under $\mathcal{H}_0$ and under no restriction ($\mathcal{H}_1$). Then

$$Q_0 = (h - w\tilde{\theta})^T \hat{\Sigma}^{-1} (h - w\tilde{\theta}) \tag{3.32}$$

$$Q_1 = (h - w\hat{\theta})^T \hat{\Sigma}^{-1} (h - w\hat{\theta}) \tag{3.33}$$

Follow Gurland et al. [118], we can define a statistic, $Q_{\mathcal{H}}$ for testing the null hypothesis $\mathcal{H}_0$ as follows:

$$Q_{\mathcal{H}} = Q_0 - Q_1 \tag{3.34}$$

It follows from the equations (3.31-3.34) that:

$$Q_{\mathcal{H}} = (B\hat{\theta} - A)^T (B(w^T \hat{\Sigma}^{-1} w)^{-1} B^T)^{-1} (B\hat{\theta} - A) \tag{3.35}$$

Gurland et al. [118] have proven that under $\mathcal{H}_0$, $Q_{\mathcal{H}}$ follows asymptotically a chi-square distribution with $r = range(B)$ degree of freedom (d.f.). When $\mathcal{H}_0$ is not true, $B\theta \neq A$; the asymptotic distribution of $Q_{\mathcal{H}}$ is a non central chi-square random variable with r d.f. and the non-centrality parameter is given by

$$\Psi = (B\theta - A)^T (B(w^T \hat{\Sigma}^{-1} w)^{-1} B^T)^{-1} (B\theta - A) \tag{3.36}$$

Thus, a test of class $\mathcal{K}_{\alpha_0}$ for (3.13) is to reject $\mathcal{H}_0$ if $Q_{\mathcal{H}} \geq \chi^2_{r,\alpha_0}$ where χ^2_{r,α_0} is the upper α_0-th percentage point of χ^2 distribution with r d.f.

3.4 Numerical Experiments

3.4.1 Model Validation

The proposed method was first validated using simulated images. For the ease of the simulation, we considered presentation attack images as printed version of the bona-fide ones. The proposed approach address only the mini-textures presenting on the surface of objects. Let omit the 3D form of the object surface. We used A. Foi et al.'s piecewise image [119], given in figure 3.3a, referred as I_{init}, to represent the surface of bona-fide object. Printed version of I_{init} represents therefore the surface of presentation attack images. The acquisition process is also simulated by employing the heteroscedastic noise model proposed in [115] and by omitting all other types of distortion. The entire flowchart of the construction of $\mathcal{H}_0$ and $\mathcal{H}_1$ simulated images is given in figure 3.2. $\mathcal{H}_0$ images are simply created by passing I_{init} image under the simulation of the acquisition process. A sample of $\mathcal{H}_0$ images is given in the figure 3.3d. For $\mathcal{H}_1$ images, we try first to simulate the distortions caused by the impression process on the initial piecewise image. Theses distortions come from different sources, as we can categorize into two groups: one caused by non-uniformity of printing paper, other caused by the specific characteristics of the

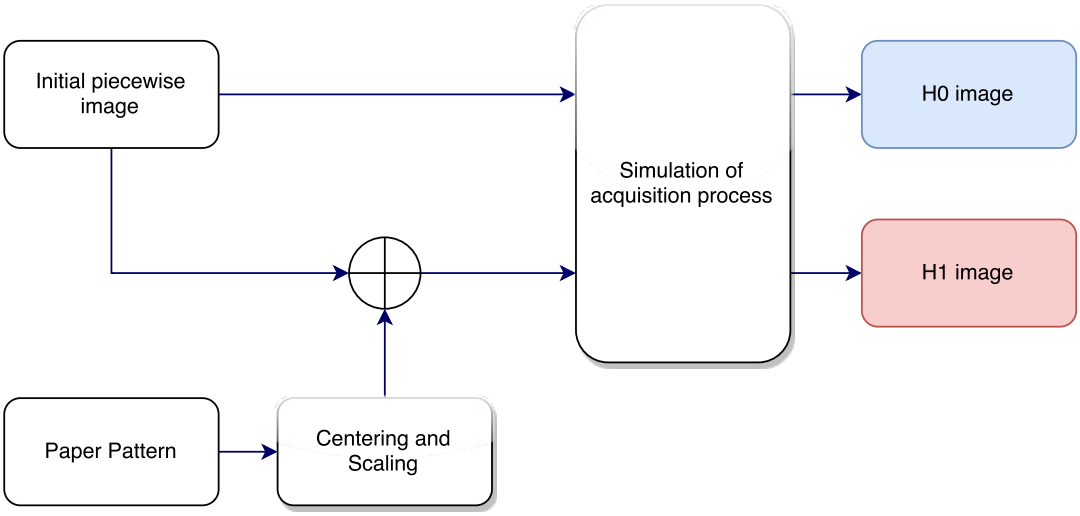


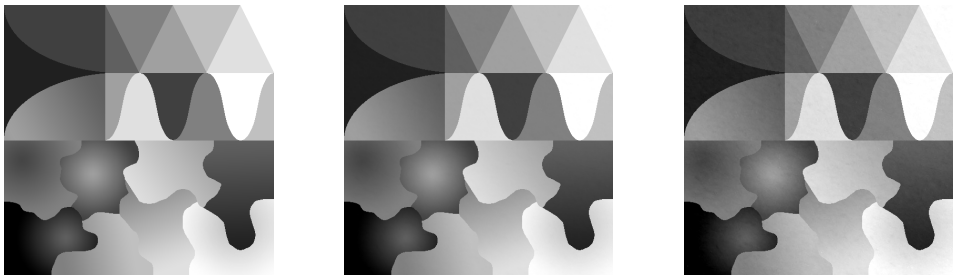
Figure 3.2: Construction flowchart of $\mathcal{H}_0$ and $\mathcal{H}_1$ simulated images

printing system (tone and color variation, halftoning process, etc.). These distortions change the statistical behavior of noise in the final image. In this simulation, we are supposed to have an ideal printing system which does not introduce any distortion. Only the non-uniformity of printing paper was taken into account. Real paper pattern, denoted I_{patt}, given in figure 3.3f, was employed to create a distorted piecewise image, denoted I_{dist}. The construction of I_{dist} is given as follow:

$$I_{dist} = I_{init} + \epsilon(I_{patt} - \bar{I_{patt}})$$

where $\bar{I_{patt}}$ is the mean of I_{patt} and ϵ is a scale factor. Figures 3.3b and 3.3c give examples of distorted piecewise image for $\epsilon = 0.2$ and $\epsilon = 0.7$ respectively. When $\epsilon = 0.2$, the distortion is very small, which is visually hard to detect. The noisy image of this one after passing through the simulation of the acquisition process is given in figure 3.3e. In our simulation, $\epsilon = 0.2$ was opted to create $\mathcal{H}_1$ images. Thousand of $\mathcal{H}_0$ simulated images were used previously to calibrate the parameters of the proposed model, given by the equation [3.12]. For the validation, we created 3000 $\mathcal{H}_0$ and 3000 $\mathcal{H}_1$ simulated images. Figure 3.4 shows the scatter plot of the couple of Gamma parameters $(1/\beta, \alpha)$, which characterizes the distribution of NLV values belonging to a given level-set, of different images. Each dot represents the couple $(1/\beta, \alpha)$ estimated from (a given level-set of) one image. Red dots correspond to $\mathcal{H}_0$ images, and blue dots correspond to $\mathcal{H}_1$ images.

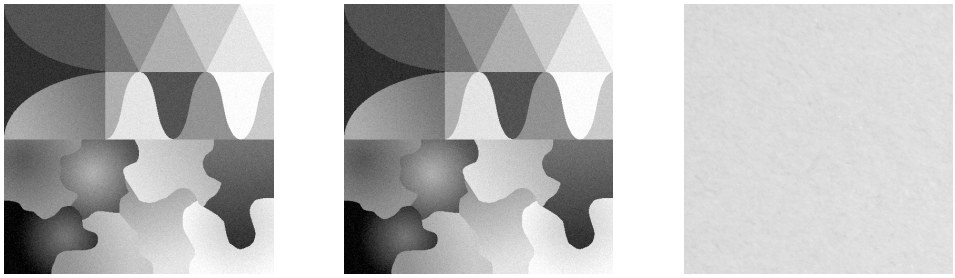
The statistic $Q_{\mathcal{H}}$ was then computed for $N = 10$ most important level-sets of each of 6000 simulated images mentioned previously. Figure 3.5 gives a comparison between the empirical distributions of $Q_{\mathcal{H}}$ under different hypotheses. Theoretical



(a) Initial piecewise image

(b) Distorted piecewise image, $\epsilon = 0.2$

(c) Distorted piecewise image, $\epsilon = 0.7$

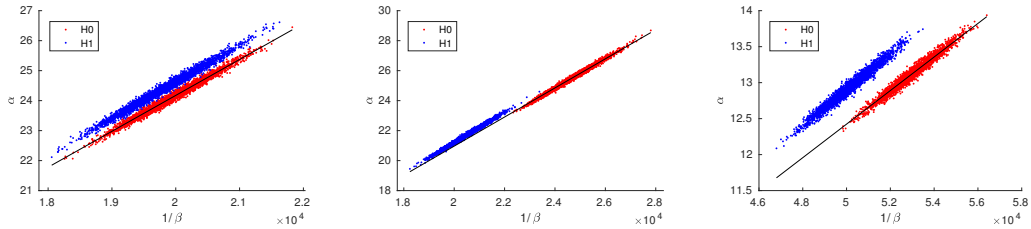


(d) Noised piecewise image

(e) Noisy image of the above one

(f) Paper pattern

Figure 3.3: Examples of simulated images.



(a) Low level-set

(b) Medium level-set

(c) High level-set

Figure 3.4: Scatter plot - Comparison between the distribution of $(1/\beta, \alpha)$ under different hypothesis for some given level-sets. Straight black presents the linear relation of these parameters under $\mathcal{H}_0$.

distribution of $Q_{\mathcal{H}}$ under $\mathcal{H}_0$ is also given. Theoretical curve match perfectly the empirical one. The discrimination between statistical distribution of the proposed statistic is clear which permit to construct a high performant detector.

3.4.2 Experimental Dataset

There are several public databases for studying face presentation attack detection such as Replay-Attack Database [120], CASIA Face Anti-Spoofing Database [121], MSU Mobile Face Spoofing Database [122], NUAA Imposter Face Database [45]. However, images or videos from these databases were taken in variable conditions. In general, the quality of the image acquisition process is very low. Actually, for

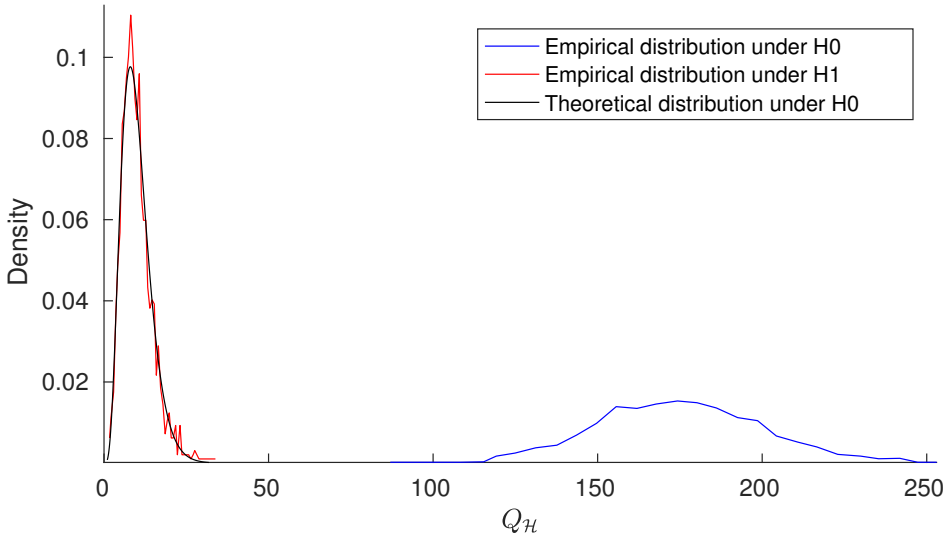


Figure 3.5: Distribution of $Q_{\mathcal{H}}$ under different hypothesis.

images of these databases, textural noise is often negligible before the acquisition noise. In the proposed method, we need to stabilize acquisition noise. Images have to be captured in the same acquisition conditions, which actually cannot be assured for existing public datasets mentioned above. We have then constructed a new database, named as Digital Forensic - Face Presentation Attack Detection (DF-FPAD) database, in our laboratory to validate our proposed approach. We constructed this database by respecting the following conditions:

- Images coming from different camera models cannot be compared in the proposed approach. In fact, each camera model has their own fingerprint [99] and different from the others' one. This difference leads to the difference in the statistical distribution of acquisition noise.

- ISO value characterizes the sensitivity of the image sensor. The higher ISO value is, the more sensitive the camera is to light and the more significant the acquisition noise is. When acquisition noise is too important, impacts caused by textural noise become less significant, the performance of the proposed solution decreases. Therefore, we proposed to keep ISO unchanged between acquisitions and not too high (400 maximum).

- Image-resolution is also an important factor in the proposed face PAD solution. When the image resolution is too small, we will not be able to model the skin-texture adequately. In this work, we focus only on high resolution image. First, we chose cameras with a high sensor-resolution to build the DF-FPAD

database. Second, the distance between camera and imaging objects had to be maintained at an adequate distance.

(a) real image **(b)** Print-Attack image **(c)** HD-print-Attack image **(d)** Projection-Attack image

Figure 3.6: Sample images of the Nikon databaset.

In this work, different types of presentation attack were considered:

- **Print-Attack**: Legitimate user's face is printed on standard A4 paper at 600dpi using a *Olivetti d-Color MF362 Plus* photocopier.
- **HDPrint-Attack**: Legitimate user's face is printed on high quality photo paper using professional photo printing service.
- **Projection-Attack**: Legitimate user's face is projected on a white board.

Different datasets have been constructed using two camera devices *Samsung Galaxy S6* and *Nikon D5200*. The two datasets are described as follows:

- **S6**: contained RAW and 3000x5328 JPEG images captured at ISO 200 by a *Samsung Galaxy S6*, using application OpenCamera. It contains 179 samples of genuine face, 161 samples of Print-Attack.
- **Nikon**: contained 4000x6000 JPEG images captured at ISO 400 by a *Nikon D5200* with the option *Noise Reduction* deactivated. The detail description of this dataset is given in the table 3.1. Some sample images from the dataset are shown in the Figure 3.6.

Description	Genuine			Print		HDPrint		Projection	
Index	1	2	3	4	5	6	7	8	9
Nb of images	500	502	377	500	499	502	377	400	500

Table 3.1: Description of the Nikon dataset

3.4.3 Results

For our experimental implementation, we opted to divide each image into $N = 100$ level-sets. The number of level-sets used to construct the statistical test, M, is configured eventually at 5, 10, 15, 20, 25. Beyond the value of 25, experimental results show that a lot of small-size level-sets were taken into account in the construction of the test; that makes its performance decrease.

RAW images

Demosaicing by interpolation was applied into RAW images (by DCRAW) before these were converted into RGB image in TIFF format. The NLV features were computed from one of the three channels of the TIFF image. In this work, we chose the green channel, which contains in most of the cases more information than the two other ones due to the configuration of the Color Filter Array (CFA). Figure 3.7 shows a typical scatter plot of gamma distribution parameters, (β^{-1}, α), estimated from a given level-set of different images. The discrimination between $\mathcal{H}_0$ and $\mathcal{H}_1$ images is visually clear, as well as, the linear relation between β^{-1} and α can be confirmed graphically. Figure 3.8 shows the Detection Error Tradeoff (DET) curves of the test with different values of M when we try to classify RAW images of the S6 dataset. The DET curve represents the relation between the Attack Presentation Classification Error Rate (APCER), also referred as the false acceptance rate, and Bona-fide Presentation Classification Error Rate (BPCER), also referred as false rejection rate. The BPCER obtained for different values of M and for a fixed value of APCER (1%) are given in the table 3.2.

M	5	10	15	20	25
RAW images	6	7	5.6	4.5	4.3
Simulated JPEG images	2.8	1.7	1.1	1	0

Table 3.2: BPCER(%) rates obtained for different values of M and for a fixed value of APCER (1%) when the detection is realized on different S6 databases.

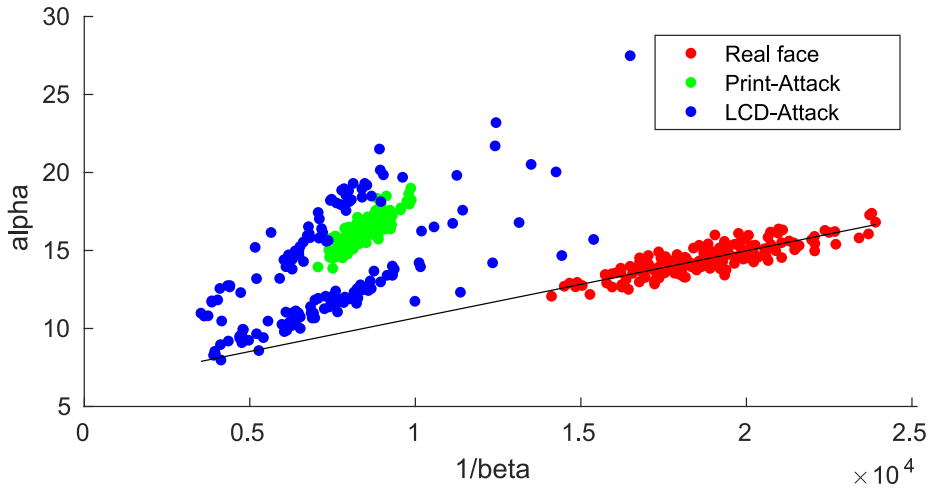


Figure 3.7: Typical scatter plot of gamma distribution parameters estimated from a given level-set of different RAW images coming from the S6 dataset; each dot represents the couple (β^{-1}, α) estimated from the given level-set of one image.

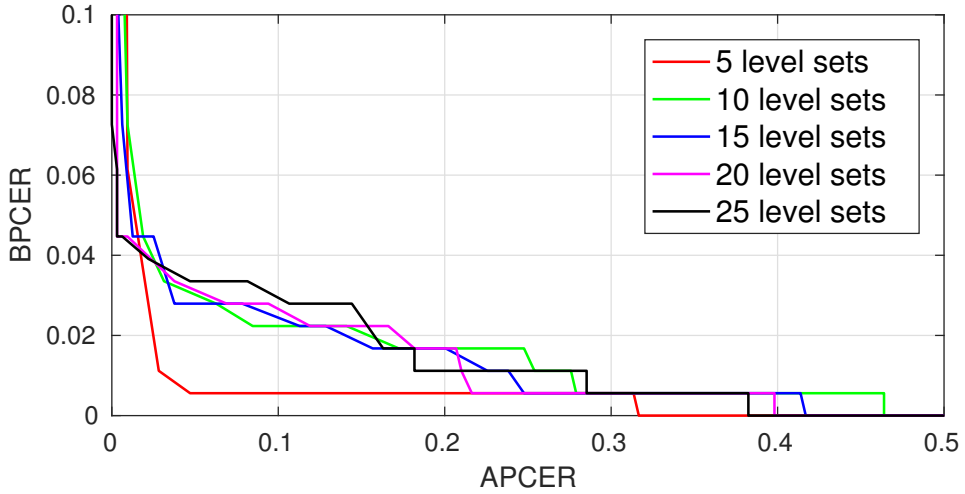


Figure 3.8: The performance of the proposed test on the S6 RAW dataset for different values of M.

Simulated JPEG images

Simulated JPEG images were developed from natural RAW images by letting them go through different principal steps of the digital image processing pipeline:

- Demosaicing
- White balancing
- Gamma Correction $(\gamma = 2.2)$
- JPEG Compression (with Quality Factor equal to 95%)

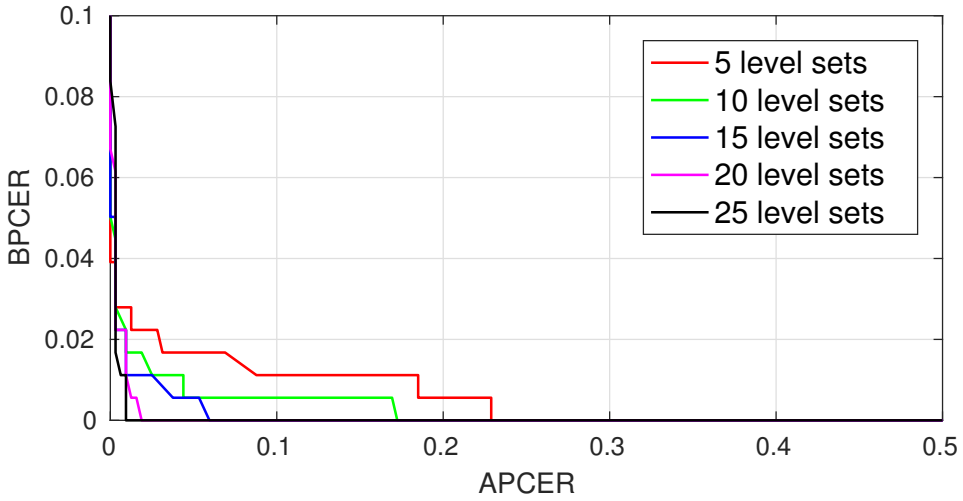


Figure 3.9: The performance of the proposed test on the simulated JPEG database created from the first RAW database for different values of M.

A simulated JPEG image database was constructed from RAW images of the S6 dataset. The performance curves of the proposed test for this database are given in Figure 3.9. It is clear to realize that the classification performance increases with the value of M. It performs even better than in the case of RAW images, see also the table 3.2 for the comparison. The discrimination between $\mathcal{H}_0$ and $\mathcal{H}_1$ simulated JPEG images may be amplified when the RAW images pass through the different operations mentioned above.

Natural JPEG images

Eventually, JPEG images, captured by the phone standard camera application, are denoised to obtain a high visual quality (less noise). This denoising operation often happened just after the white balancing and just before the gamma correction operation in the digital image processing pipeline. Each phone's constructor has their private implementation for this denoising. This denoising operation unfortunately destroys the favorable high-frequency information exploited in our proposed method. Consequently, for the classification of JPEG images of the S6 dataset, the proposed approach performed badly. However, we can deactivate this denoising operation. For recent Android smartphones, including our Samsung Galaxy S6, by using the Camera2 API, we can build applications which give us the possibility to configure manually all camera settings and apparently including the implementation of the mentioned denoising operation. For a digital camera, such as Nikon D5200, we have options to deactivate implemented denoising operations. Images from the

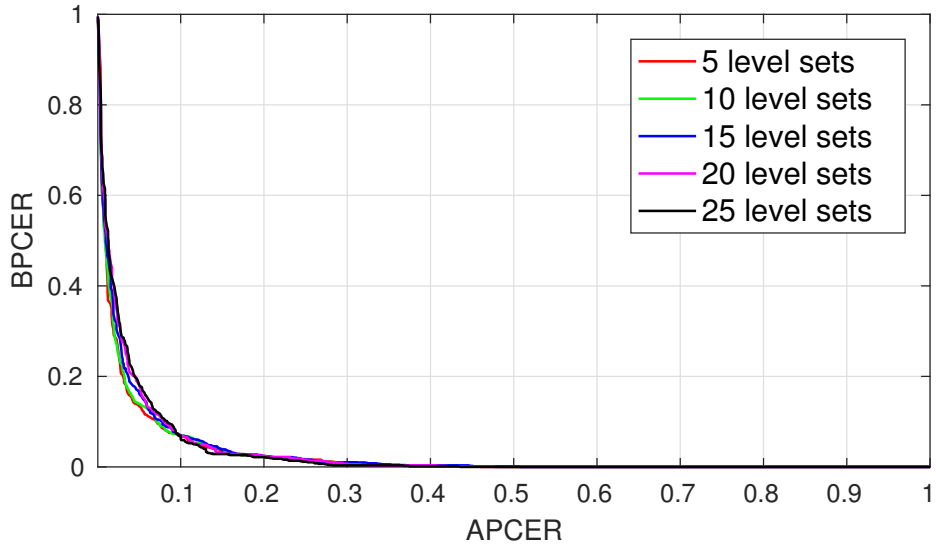


Figure 3.10: The performance of the proposed test on the Nikon database for different values of M.

Nikon dataset were taken with these options deactivated. Figure 3.10 shows the performance of the proposed test for the classification of these images. The classification performance is stable for different values of M. The proposed approach has been also compared with the reference methods proposed in [46] and [8] on the Nikon dataset. In our approach, we need only bona-fide images to estimate the parameters of the test. We used 500 bona-fide face images for this estimation. For the two reference approaches, which are non-parametric classification solutions, presentation attack images are needed to train the SVM models. In the Nikon dataset, attack images were divided into three subsets basing on the type of attack: Print-Attack, HDPrint-Attack, and Projection-Attack. Each subset was divided into two parts, one of these parts would be used to train the SVM models of reference approaches. The proposed approach was compared with the reference ones in different situations. When all types of face presentation attack are supposed to be known by anti-spoofing systems, reference approaches performed better than ours, as we can see in Fig. 3.11.

However, when the systems have previously unseen presentation attack instrument (PAI), which is usually the case in practice, it is proven that reference approaches would classify imprecisely images from a new type of attack, their performance dropped dramatically. Meanwhile, for our solution, which does not require prior knowledge of attack images, unseen PAI does not bother its performance. Figure 3.12 shows the performance comparison between our proposed solution and the reference ones under different assumptions about the unseen PAI on different types of face presentation attack. Our solution performed steadily and obviously better

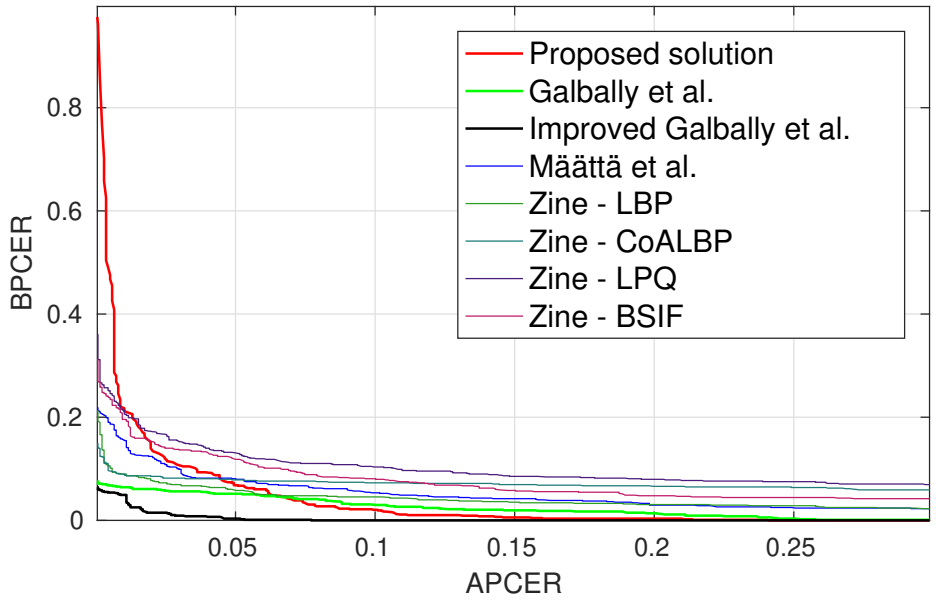
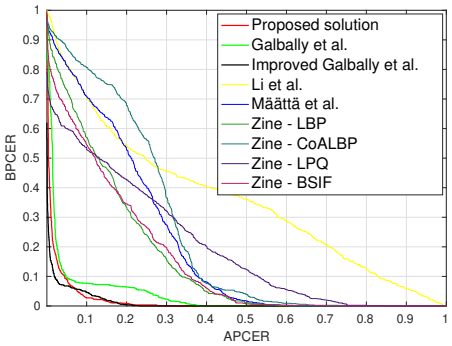
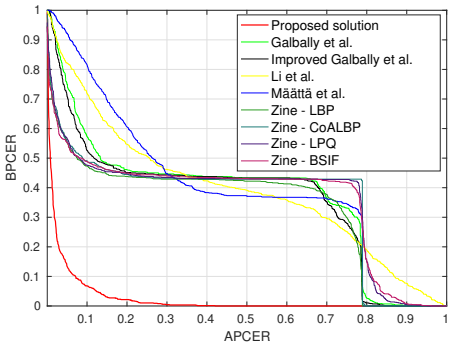


Figure 3.11: Performance of the proposed solution and the reference approaches when all types of spoofing attack are supposed to be known.

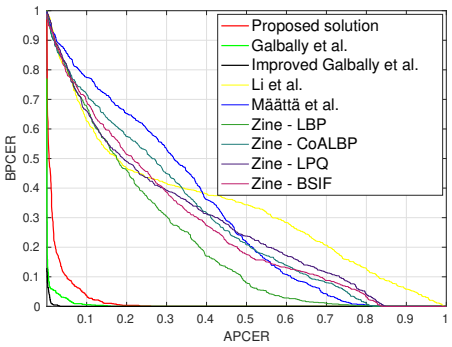
than the two reference ones. Widely-used Half Total Error Rate (HTER) is also used in our study as the evaluation parameter to compare the performance of the proposed solution with the reference ones. HTER is half of the sum of the Attack Presentation Classification Error Rate (APCER) and the Bona-fide Presentation Classification Error Rate (BPCER), evaluated at the threshold τ_{EER}. By definition, τ_{EER} is the threshold which permits to have an Equal Error Rate (EER), which means APCER=BPCER=EER, when the method is evaluated with the training dataset. The Table 3.3 shows the results obtained for different methods when different configurations of the training dataset is proposed. Our proposed solution performs steadily and, for some configurations of the training dataset, it performs better than the other ones. Galbally et al.'s solution [41] performs in general better than ours. However, by considering the proposed statistics $Q_{\mathcal{H}}$ (Equation 3.34) as an Image Quality Assessment, we have added $Q_{\mathcal{H}}$ as an additional measure into the set of 14 measures proposed in [41]. The presence of $Q_{\mathcal{H}}$ as a descriptor improves considerably the classification performance, see Figure3.11. The improved version of Galbally et al.'s approach outperforms all the other ones.



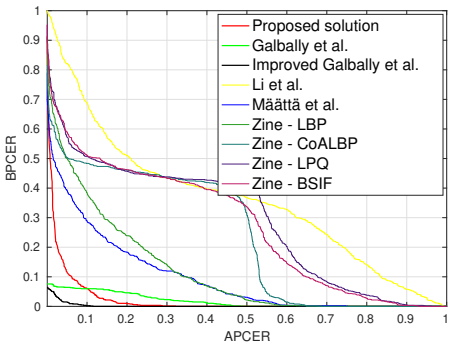
(a) Only Print-Attack is known



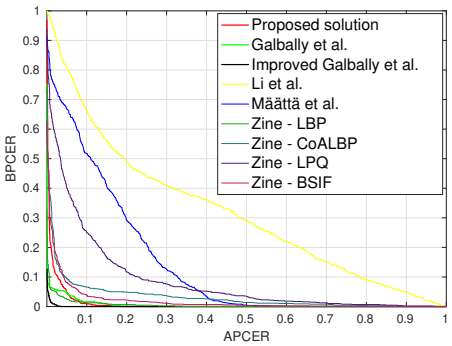
(b) Only Projection-Attack is known



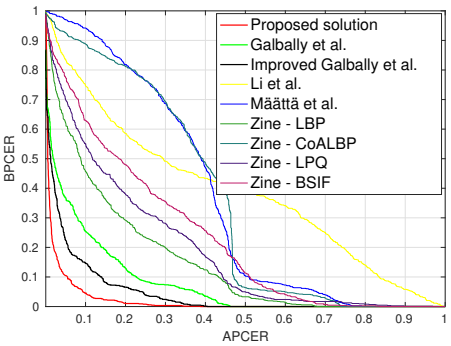
(c) Only HDPrint-Attack is known



(d) Projection and HDPrint Attack are known



(e) Print and HDPrint Attack are known



(f) Projection and Print Attack are known

Figure 3.12: Performance of the proposed solution and the reference approaches under different configurations of training dataset. Once a type of attack is mentioned as known, the SVM models of reference approach have been trained with haft of images from this type of attack. DET curves were obtained by evaluating the trained models with images which were not used for training. Our solution is compared with solutions proposed by Galbally et al. [41], Li et al. [44], Määttä et al. [46], Boulkenafet et al. [8]. Different performance curves given by the Boulkenafet et al.'s approach are labeled with the prefix *Zine-*.

Training sets	**1 4**	**1 6**	**1 8**	**1 4 6**	**1 4 8**	**1 6 8**	**1 4 6 8**
Proposed sol.	17.9	15.7	**12.4**	16.8	15.5	14.9	15.6
Improved Galbally	**11.3**	**2.9**	52	**3.6**	**14.3**	**3.3**	**3.7**
Galbally et al. [41]	15.6	3.5	53.6	3.9	17.9	4.1	4.2
Li et al [44]	61	66.1	54.1	65.7	57.3	64.4	62.7
Määttä et al. [46]	24.2	41.9	54.4	24	38.2	21.2	7.5
Zine- LBP[8]	22.4	32.6	52.1	3.4	26.7	21.9	4.9
Zine-CoALBP[8]	24.4	42	59.4	7.2	35.6	42.4	12.3
Zine-LPQ [8]	35.5	42.1	59.5	18.5	29.1	43.9	16.7
Zine-BSIF [8]	23.1	41.8	58.7	7.5	33.1	40.3	15.4

Table 3.3: Face PAD performance evaluated by HTER (%) for different configurations of training dataset; see Table 3.1 for the indexation of the training sets.

3.5 Conclusion

In this section, we have introduced a new method for detecting face presentation attack based on the statistical behavior of image noise. We considered that mini-texture information presenting on the surface of imaging objects, which can be interpreted as textural noise in images, can provide us interesting information to distinguish different surfaces. Furthermore, the surface of bona-fide presentation and attack presentation of facial images differs generally one from the other. In this work, we tried to model this difference to propose a face presentation attack detection solution. Textural noise is technically inseparable from the acquisition noise. We can only study this noise by considering the combination of the two noise sources. The distribution of this combination is quite really hard to model. We decided therefore to study the difference between noise distributions by observing indirectly the behavior of noise variance. In order to model appropriately the statistical behavior of textural noise variance, we assume that the acquisition noise has a fixed variance. To satisfy this assumption, we have to stabilize the statistical behavior acquisition noise. In this work, only images from a same camera device with a fixed configuration of ISO are able to be compared one from the other. By fixing the acquisition pipeline, we expect to have a stabilized acquisition noise. However, there always exists some uncontrollable factors which affect the acquisition noise, such as the lighting conditions. These factors induct in someway the change in the variance of acquisition noise and indirectly affect the performance of the proposed face PAD solution. Textural noise presenting on the skin zone of bona-fide facial images may also differs one from the other. Due to race, age and skin healthy, different people do not generally have a same texture on their face. Therefore, we cannot propose a same model for bona-fide images of different people. The proposed solution is then

appropriated for applications which have a mono-user authentication system, such as management application for personal bank account. The distance between imaging object and camera system must also be considered, because it affects directly on the resolution of textural noise over the skin zone. The proposed solution relies directly on the behavior of textural noise of facial image to detect face presentation attack. It is quite sensitive with the aging, makeup, etc. In this work, we omitted all these nuisance factors. However, in practice, these factors have to be taken in account. It can be, for example, proposed to calibrate occasionally the system to deal with aging problem or to stock additionally the model of some special bona-fide cases to deal with makeup. We remarked that the proposed solution can permit us, in general, to differentiate a bona-fide surface from its falsified version, a version created by digitalizing and reprinting the bone-fide one. This remark gave us an idea about creating something that can protect printed document from illegal duplication. Finally, a watermarking technique that help to secure printed code-barre has been proposed. The contribution of the work is presented in the following chapter.

Chapter 4

A watermarking technique to secure printed matrix bar-code

Contents

4.1 Introduction

Nowadays, counterfeiting of consumer goods is becoming an ongoing concern for the consumers, enterprises and the whole society. We can discover counterfeit goods in many domains, such as toys, food, textile, medication, etc. Consumers are the first victims. They receive poor-quality goods at an excessive price, and these products may threaten their health and safety. The circulation of these products also involves social costs and the losses of unpaid tax for governments. For legitimate-brand companies, they lost a part of their market. Consumers' bad experiences caused by counterfeit products can also endanger the reputation and value of the brand.

Companies have to have more awareness for fighting against counterfeiters to protect themselves and their consumers. They have to make sure that their trademark is adequately protected against the menace. A myriad of technologies [123], such as holograms [124], RFID [125, 126, 127] or NFC tags [128, 129], biometric markers or inks [130],... are proposed to preserve and certify the authenticity of products. These solutions vary considerably in their sophistication and cost. For low-cost products, such as medicines or textiles, companies do not have much more willing to pay for an expensive RFID tag or hologram. A cheaper solution would be more appreciated in this case.

Matrix barcodes is a cheap and effective solution to embed the identification or tracking information of products. However, they cannot be used as a security element for fighting counterfeiting. These codes are easy to be copied or regenerated. We can reproduce them without any difficulty from the message embedded, or we can create a fake one by scanning and reprinting or simply photocopying an original printed code. Therefore, practically, we cannot use standard printed matrix barcodes to verify the authenticity of products. However, we can improve the standard ones to make them hard to be copied or regenerated, and so a solution to authenticate products.

In this chapter, we introduced a novel watermarking technique, which embeds a particular random micro-texture into a matrix barcode and transforms this one into a security layer for making packaging anti-counterfeited. The printed secured matrix barcode is hard to be reproduced by counterfeiters. In fact, any degradations caused by the counterfeiting process will change the statistical behaviors of the embedded micro-texture. Statistical detectors basing on the hypothesis testing framework are also introduced to classify authentic and counterfeited printed barcodes.

Experimental results confirm the usability and the effectiveness of the proposed anti-counterfeiting solution.

For the ease of implementation, we focused our work on QR [131] (Quick Response) codes. Therefore, the proposed anti-counterfeiting solution is named as W-QR in the following section. However, we can also apply the proposed watermarking technique to secure any other types of matrix barcode.

4.2 Description of W-QR code

4.2.1 Concept

The application of QR codes on physical support implies a printing process. The reading of the codes implies after that a scanning process. Both these mentioned process add distortions to images. Therefore, the image treated by the reading/authentication system is a degraded version of the original digital code. A standard QR code is basically used to encode and transmit information. By its conception, the distortions caused by the P&S process, considered as noise, does not disturb the reading of the information transmitted. However, in authentication scenarios, these distortions can be interested, because of the physically unclonable characteristics of the P&S process [67].

Denote I_o and I respectively the original numerical code and the scanned image of the printed code. Assume that we can manage to model and stabilize the P&S process, denote $f_\theta^{PS}(.)$ the stochastic function which transforms I_o to I:

$$I = f_\theta^{PS}(I_o) \tag{4.1}$$

where θ is a set of unknown parameters which characterizes the stochastic behavior of the related P&S process. For a given printing system, when the scanning/acquisition system is fixed, the value of θ is assumed to be constant. The function $f_\theta^{PS}(.)$ is physically unclonable. In practice, it's is extremely hard to characterize this function. Its inverse function is therefore much more difficult to be figured out. By knowing I, it's is hard to retrieve I_o. Inversely, it's also difficult to obtain a similar I by having I_o, unless you have the same printing system which is characterized by $f_\theta^{PS}(.)$.

When I_o is a standard QR code, which simply consists of black squares arranged in a square grid on a white background (see Figure 4.1), even when the function

Figure 4.1: Sample QR barcode.

$f_\theta^{PS}(.)$ and its inverse are unknown, we can easily retrieve I_o basing on the information encoded.

In order to exploit the P&S process as a key element to develop secured QR codes, we proposed to modify I_o to make it complicated so that retrieving I_o from I requires a certain knowledge of the employed P&S process, not only the information encoded. In fact, we proposed to substitute the black and/or white modules of the standard codes by a specific texture. The texture embedded is configured so that they change the behavior of the codes under the P&S process but they do not disturb the normal reading of the information encoded.

The figure 4.2 describes the general construction flowchart for our proposed W-QR code. Firstly, binary QR codes, which encodes the product public information, are generated by standard generation algorithms. We generate then a texture which is characterized by a private key or by some secret setting parameters. After that, we combine the generated texture with the binary QR code by substitution to construct W-QR code. The reading process is simple, described by the flowchart given in the

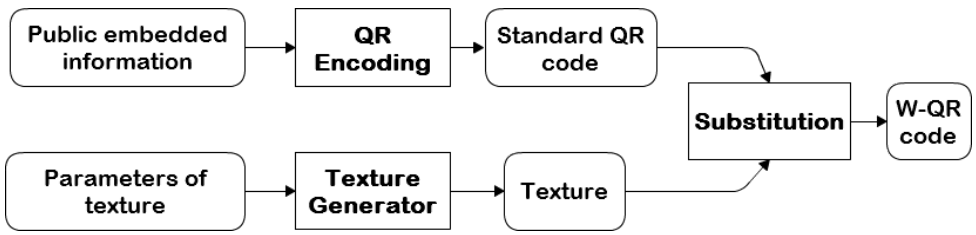


Figure 4.2: Proposed flowchart for the construction of W-QR.

figure 4.3. Public information encoded by the QR code is decoded by standard reading application. By regenerating the original standard QR code from the information decoded, we obtain a mask which separates black modules and the background of the scanned code. The embedded texture is then extracted. Finally, by studying the

characteristics of the texture, we manage to make a decision about the authenticity of scanned code.

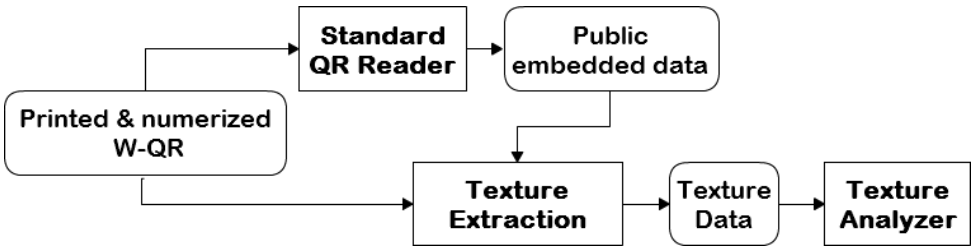


Figure 4.3: Proposed flowchart for the reading and authentication of W-QR.

4.2.2 Clipping Gaussian Noise Texture

The key factor in the proposed concept is the embedded texture and the measures which permit to verify its authenticity. There may be several options for that. For example, Dirik et al. [132] proposed to substitute the black module of the QR code by a type of copy detection pattern, the verification method was also provided. In this chapter, we proposed to substitute the white background by a novel type of texture, named as Clipping Gaussian Noise (CGN). The proposed texture is described in the following paragraphs. The verification method is proposed in the next sections.

The CGN texture is created from an image of a 2D random Gaussian signal by replacing all values which are out of a predefined interval, denoted as clipping interval, by the interval boundary. In the case where the value of a pixel of an image is an integer encoded by 8 bits, its value varies from 0 to 255. A 2D random Gaussian signal, which is quantized to be an integer signal, takes its values over all the set of integers. By defining the clipping interval as $[0, 255]$, the texture is created by replacing all the values which are higher than 255 by 255, and all the ones which are smaller than 0 by 0. The replacement creates an artificial clipping effect, which produces a texture saturated in the bright-rank, in the dark-rank or in both of the two, depending on the parameters which configures the random signal. We can characterize the CGN texture by the mean μ and the standard deviation σ of the Gaussian random signal. Let denote $T_{\mu,\sigma}$ as the obtained texture. Figure 4.4 gives an illustration for the idea about the CGN texture; the texture presented in the figure is $T_{200,70}$; on the left, it is the intensity histogram of the texture.

In order to take in account the overall intensity and the visibility of the texture, it is proposed to scale down the previous signal by a factor ϵ $(\epsilon < 1)$ and centered

it around a value μ_0. So that, the final numeric definition of the proposed texture should be given as follows:

$$I_o = \mu_0 + \epsilon(T_{\mu,\sigma} - E(T_{\mu,\sigma})) \tag{4.2}$$

where $E(.)$ denotes the average operation.

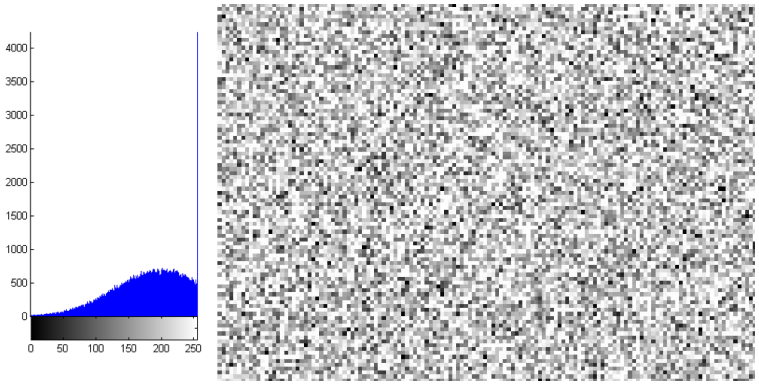


Figure 4.4: CGN (right to left): the texture and its histogram.

We opted to substitute only the white background of the code by the proposed texture. The black modules are still black. Hence, the authenticity of the W-QR codes is decided from only the textured background. Figure 4.5 shows the different versions of the proposed W-QR codes.

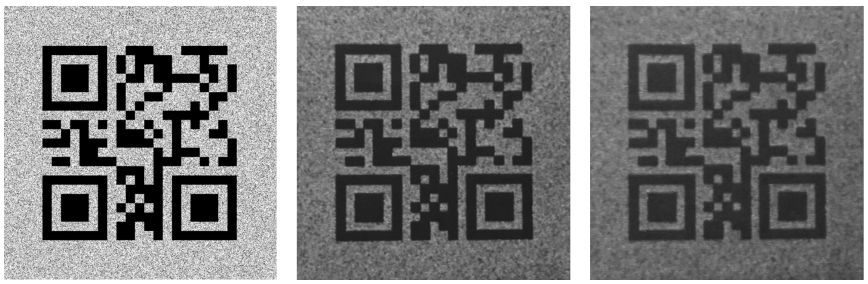


Figure 4.5: Different versions of W-QR (left to right): numeric version, printed genuine one, fake one created by printing the scanned version of the second one.

4.3 Statistical Modeling

4.3.1 Variance Model

Let $\mathcal{Z} = \{z_i | i \in \{1,2,...,n\}\}$ is a sample of an independently and identically distributed random variable with the standard deviation σ_0. Considering S the estimate

of the sample variance, we have:

$$S = \frac{1}{n-1}\sum_{i=1}^{n} s_i \tag{4.3}$$

where

$$s_i = (z_i - \bar{z})^2 \tag{4.4}$$

and $\bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i$. We obtain the following theorem demonstrated in Annexe A.2.

Theorem 2 *For $\mathcal{Z} = \{z_i | i \in \{1,2,...,n\}\}$ is a sample of an independently and identically distributed random variable with the standard deviation σ_0. We have*

$$s_i \rightarrow \Gamma(\frac{1}{2}, 2\frac{n-1}{n}\sigma_0^2) \tag{4.5}$$

and

$$S \rightarrow \Gamma(\frac{n}{2}, \frac{2\sigma_0^2}{n}) \tag{4.6}$$

where $\Gamma(\alpha,\beta)$ is the Gamma distribution, of which the probability density function is defined as follows:

$$f(x) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}} x^{\alpha-1} e^{-\frac{x}{\beta}}, \text{ where } x \in (0,\infty). \tag{4.7}$$

4.3.2 Statistical Properties of W-QR Images in DCT domain

We divide an image into non-overlapping blocks and study each one in the frequency domain. In our work, the Discret Cosine Transform (DCT) is employed to transform spatial images into a frequency presentation. We plan to work with JPEG format images. Since JPEG compression process over 8×8 non-overlapping blocks, to assure the independence between the blocks; the size 8×8 is opted in our study. Denote $\mathcal{I}$ the set of block index, and z_{ji}, $i \in \mathcal{I}$, $j \in \{1,2,..,64\}$, the $j-$th coefficient in the DCT domain of the i-th block. Denote $\mathcal{Z}_j = \{z_{ji} | i \in \mathcal{I}\}$, the set of DCT coefficients within the j-th subband of all considered blocks. Prior researches [133, 134] proved that the DCT coefficients within a sub-band can be considered as a sample of an i.i.d random variable. It follows that $\mathcal{Z}_j$ is a sample of an i.i.d random variable. For printed W-QR images, the texture-embedded zone contains statistically the same information in their original digital version. They also pass through the same printing and acquisition process, which are assumed to be stable. The distortions caused by these processes are then probably similar between images. Consequently,

the statistical behavior of these images in frequency domain should be similar. In other words, in the DCT domain, the coefficients within a given subband should follow the same distribution for all images. Falsified printed W-QR images, which were created for example by scanning genuine printed W-QR codes and reprinting, should be affected by several distortions, that make them different from images of the genuine one. This difference can be observed in the frequency domain. By studying the statistical behavior of DCT coefficients, we can manage to detect falsified W-QR codes from genuine ones. Studying directly the distribution of DCT coefficients may be difficult. Under the assumption that $\mathcal{Z}_j$ is a sample of an i.i.d random variable, it is proposed to use the proposed Variance Model, presented in the previous section, to simplify our study. For the j-th DCT subband, let divide $\mathcal{Z}_j$ into n-length subsamples. Denote $X_j = \{x_{ji} | i \in \mathcal{I}\}$, where x_{ji} are defined as follows:

$$x_{ji} = (z_{ji} - \bar{z}_{ji})^2, \forall i \in \mathcal{I} \tag{4.8}$$

where $\bar{z}_{ji}$ is the average value of the subsample which contains z_{ji}. It follows by the Theorem 2 that:

$$x_i \sim \Gamma\left(\frac{1}{2}, (n-1)b_j\right) \tag{4.9}$$

where b_j is a constant which characterizes the sample j, it can be defined in function of the variance of the distribution charaterizing $\mathcal{Z}_j$. In the DCT domain, we have in total one DC subband and 63 AC subbands. The difference between genuine and falsified images is more significant in the low-level AC subbands. We may opt to exploit the subband where the difference is the most significant to build a detector using the hypothesis testing framework. One-subband-sample-based detectors are proposed in detail in section 4.4. We can also accumulate the difference from sample of more than one subbands. Section 4.5 proposes a strategy to combine the statistics from these samples to build a detector.

4.4 One-subband-sample-based (OSSB) Detector

We can use sample from only a given DCT subband to build statistical test for the propoped detection problem. By omitting the index of subband, X_j is rewritten as follows $X = \{x_i\}_{i \in \mathcal{I}}$, j here is the index of the given DCT subband. The goal of the

test is to decide between two hypotheses defined for $\forall i \in \mathcal{I}$ as follows:

$$\begin{cases} \mathcal{H}_0 : \{x_i \sim \Gamma(a, b_0)\} \\ \mathcal{H}_1 : \{x_i \sim \Gamma(a, b_1)\}, b_1 \neq b_0 \end{cases} \tag{4.10}$$

where $a = \frac{1}{2}$, b_0 is known, b_1 can be known or unknown. In this work, we focus on designing a test that allows us to guarantee a prescribed false alarm probability. Let

$$\mathcal{K}_{\alpha_0} = \{\delta : \mathbb{P}_{\mathcal{H}_0}[\delta(X) = \mathcal{H}_1] \leq \alpha_0\} \tag{4.11}$$

be the class of tests with a false alarm probability upper-bounded by α_0. Here $\mathbb{P}_{\mathcal{H}_h[E]}$ stands for the probability of event E under hypothesis $\mathcal{H}_h$, $h \in \{0,1\}$. Assuming that b_1 is known, a Likelihood Ratio Test is given. We also proposed a Generalized Likelihood Ratio Test which covers the lack of knowledge of b_1.

4.4.1 Likelihood Ratio Test (LRT) for Two Simple Hypotheses

In virtue of the Neyman-Pearson lemma [135], the most powerful test δ solving the problem 4.10 is the Likelihood Ratio Test given by the following decision rule:

$$\delta(X) = \begin{cases} \mathcal{H}_0 \text{ if } \Lambda(X) = \sum_{i \in \mathcal{I}} \Lambda(x_i) < \tau \\ \mathcal{H}_1 \text{ if } \Lambda(X) = \sum_{i \in \mathcal{I}} \Lambda(x_i) \geq \tau \end{cases} \tag{4.12}$$

where τ is the solution of the following equation:

$$\mathbb{P}_{\mathcal{H}_0}[\Lambda(X) \geq \tau] = \alpha_0 \tag{4.13}$$

and $\Lambda(x_i)$, the log-likelihood ratio of the observation x_i, is given as follow:

$$\Lambda(x_i) = log\frac{\mathcal{L}_1(x_i)}{\mathcal{L}_0(x_i)} = a \times log\frac{b_0}{b_1} + x_i \times (\frac{1}{b_0} - \frac{1}{b_1}) \tag{4.14}$$

where $\mathcal{L}_h(x_i), h \in \{0,1\}$ is the likelihood function for the observation x_i under hypothesis $\mathcal{H}_h$, given by:

$$\mathcal{L}_h(x_i) = \frac{1}{\Gamma(a) b_h^a} x_i^{a-1} exp(-\frac{x_i}{b_h})$$

So that,

$$\Lambda(X) = aNlog\frac{b_0}{b_1} + (\frac{1}{b_0} - \frac{1}{b_1})\sum_{i\in\mathcal{I}} x_i \tag{4.15}$$

where N is the size of $\mathcal{I}$. The first term of $\Lambda(X)$ is constant, which could be omitted. Let define $\Lambda_1(X) = (\frac{1}{b_0} - \frac{1}{b_1})\sum_{i\in\mathcal{I}} x_i$. Under hypothesis $\mathcal{H}_h, h \in \{0,1\}$, by employing the Central Limit Problem, we have that:

$$\Lambda_1(X) \sim \mathcal{N}(m_h, v_h) \tag{4.16}$$

where

$$m_h = Nab_h\frac{b_1 - b_0}{b_0 b_1} \tag{4.17}$$

$$v_h = Nab_h^2\frac{(b_1 - b_0)^2}{b_0^2 b_1^2} \tag{4.18}$$

Let define

$$\Lambda^\star(X) = \frac{\Lambda_1(X) - m_0}{\sqrt{v_0}} \tag{4.19}$$

the rule 4.12 can be rewritten as follows:

$$\delta^\star(X) = \begin{cases} \mathcal{H}_0 \text{ if } \Lambda^\star(X) < \tau^\star \\ \mathcal{H}_1 \text{ if } \Lambda^\star(X) \geq \tau^\star \end{cases} \tag{4.20}$$

where $\tau^\star$ is a constant defined mathematically in function of the prescribed false-alarm probability. The decision threshold $\tau^\star$ and the power function $\beta_{\delta^\star}$ are given in the following theorem:

Theorem 3 *When b_h, $h \in \{0,1\}$, are known, to obtain the prescribed false-alarm probability α_0, the decision threshold and the power function of the test $\delta^\star$ are given by:*

$$\tau^\star = \Phi^{-1}(1 - \alpha_0) \tag{4.21}$$

$$\beta_{\delta^\star} = 1 - \Phi(\frac{m_0 - m_1 + \tau^\star\sqrt{v_0}}{\sqrt{v_1}}) \tag{4.22}$$

where $\Phi(.)$ and Φ^{-1} denote respectively the cumulative distribution function of the standard Gaussian random variable and its inverse.

4.4.2 Generalized Likelihood Ratio Test (GLRT)

When b_1 is unknown, the problem now is to verify whether the inspected texture image is belong to the defined $\mathcal{H}_0$ texture family. By replacing b_1 in the equation 4.14 by $\hat{b}$, a consistent estimate of the scale parameter of the Gamma distribution, we obtain:

$$\hat{\Lambda}(x_i) = a \times log\frac{b_0}{\hat{b}} + x_i \times (\frac{1}{b_0} - \frac{1}{\hat{b}}) \tag{4.23}$$

where

$$\hat{b} = \frac{1}{Na}\sum_{i\in\mathcal{I}} x_i \tag{4.24}$$

So that,

$$\hat{\Lambda}(X) = \sum_{i\in\mathcal{I}} \hat{\Lambda}(x_i) = \frac{1}{b_0}\sum_{i\in\mathcal{I}} x_i - Na \times log(\frac{1}{N}\sum_{i\in\mathcal{I}} x_i) + C \tag{4.25}$$

where $C = Nalog(b_0/a)$ is a constant which will be omitted in the following development. We can rewrite the equation 4.25 by omitting the constant C as follows:

$$\hat{\Lambda}(X) = \frac{N}{b_0}\left[\bar{X} - ab_0 log(\bar{X})\right] \tag{4.26}$$

where $\bar{X}$ is the mean of the sample X, given by:

$$\bar{X} = \frac{1}{N}\sum_{i\in\mathcal{I}} x_i$$

Under hypothesis $\mathcal{H}_h, h \in \{0,1\}$, by invoking the Central Limit Theorem, we have that:

$$\bar{X} \to \mathcal{N}(\mu_h, \sigma_h^2) \tag{4.27}$$

where

$$\sigma_h = \sqrt{\frac{a}{N}} b_h \quad \text{and} \quad \mu_k = ab_h \tag{4.28}$$

Let $\bar{X}_h = \frac{\bar{X}-\mu_h}{\sigma_h}$, we have that:

$$\bar{X}_h \to \mathcal{N}(0,1) \tag{4.29}$$

and under hypothesis $\mathcal{H}_k$, it follows that:

$$\hat{\Lambda}(X) = \frac{N}{b_0}\left[\sigma_h X^\star + \mu_h - \hat{\mu}_0 log(\mu_h)\right] - \frac{N}{b_0}\mu_0 log(\frac{\sigma_h}{\mu_h}X^\star + 1) \tag{4.30}$$

Taylor's theorem gives that:

$$log(\frac{\sigma_h}{\mu_h}X^\star + 1) \approx \frac{\sigma_h}{\mu_h}X^\star - \frac{\sigma_h^2}{2\mu_h^2}\bar{X}_h^2 \tag{4.31}$$

By combining the equations 4.30, 4.31 and 4.28, we obtain that, under hypothesis $\mathcal{H}_h$:

$$\hat{\Lambda}(X) \approx \frac{1}{2}\left(\bar{X}_h + c_h\right)^2 + d_h, \tag{4.32}$$

where

$$c_h = \sqrt{Na}\frac{b_h - b_0}{b_0}, \tag{4.33}$$

$$d_h = Na\frac{b_h - b_0 log(ab_h)}{b_0} - \frac{1}{2}c_h^2. \tag{4.34}$$

Particularly, we have that $c_0 = 0$. Let define:

$$\hat{\Lambda}^\star(X) = 2\left(\hat{\Lambda}(X) - d_0\right). \tag{4.35}$$

Then, under hypothesis $\mathcal{H}_0$ we have that $\hat{\Lambda}^\star(X) \approx \bar{X}_0^2$, which behaves like a chi-square random variable with one d.f. So that, we have the following theorem:

Theorem 4 *Under the hypothesis $\mathcal{H}_0$:*

$$\hat{\Lambda}^\star(X) \to \chi^2(1) \tag{4.36}$$

Following Neyman-Peason lemma, the most powerful test $\hat{\delta}$ solving 4.10 is the Generalized Likelihood Ratio Test given by the following decision rule:

$$\hat{\delta}(X) = \begin{cases} \mathcal{H}_0 \text{ if } \hat{\Lambda}^\star(X) < \hat{\tau} \\ \mathcal{H}_1 \text{ if } \hat{\Lambda}^\star(X) \geq \hat{\tau} \end{cases} \tag{4.37}$$

where, again to ensure $\hat{\delta}$ to be in the class $\mathcal{K}_{\alpha_0}$, $\hat{\tau}$ is the solution of the equation

$$\mathbb{P}_{\mathcal{H}_0}\left(\hat{\Lambda}^\star(X) \geq \hat{\tau}\right) = \alpha_0 \tag{4.38}$$

Theorem 5 *The threshold decision $\hat{\tau}$ and the power function $\beta_{\hat{\delta}}$ of the test $\hat{\delta}$ are given as follows:*

$$\hat{\tau} = \Phi^{-1}_{\chi_1^2}(1 - \alpha_0), \tag{4.39}$$

$$\beta_{\hat{\delta}} = 1 - \Phi_{\chi^2_{1,\lambda}}\left(\hat{\tau} + 2(d_0 - d_1)\right), \tag{4.40}$$

where $\Phi^{-1}_{\chi^2_1}(.)$ *denotes the inverse cumulative distribution function of a chi-square random variable with one d.f;* $\Phi_{\chi^2_{1,\lambda}}(.)$ *denotes the cumulative distribution function of a non-central chi-square random variable with one d.f and the non-centrality parameter* $\lambda = c_1^2$.

4.5 Multiple-subbands-sample-based (MSSB) Detector

When more than one subbands are taken in account in the decision process, denote $\mathcal{I}_{sub}$ the set of interested subbands. Our goal is now to decide between two hypotheses defined for $\forall i \in \mathcal{I}, \forall j \in \mathcal{I}_{sub}$ as follows:

$$\begin{cases} \mathcal{H}_0 : \{x_{ji} \sim \Gamma(a, b_{j0})\} \\ \mathcal{H}_1 : \{x_{ji} \sim \Gamma(a, b_{j1})\}, b_{j1} \neq b_{j0} \end{cases} \tag{4.41}$$

We have:

$$\mathbb{P}(X_{j,\forall j\in\mathcal{I}_{sub}}|\mathcal{H}_k) = \prod_{j\in\mathcal{I}_{sub}}\prod_{i\in\mathcal{I}}\mathbb{P}(x_{ji}|\mathcal{H}_k) \tag{4.42}$$

where $k \in \{0, 1\}$. The log-likelihood ratio is then defined as follows:

$$\Lambda(X_{j,\forall j\in\mathcal{I}_{sub}}) = log\frac{\mathbb{P}(X_{j,\forall j\in\mathcal{I}_{sub}}|\mathcal{H}_1)}{\mathbb{P}(X_{j,\forall j\in\mathcal{I}_{sub}}|\mathcal{H}_0)}. \tag{4.43}$$

So that,

$$\Lambda(X_{j,\forall j\in\mathcal{I}_{sub}}) = \sum_{j\in\mathcal{I}_{sub}}\sum_{i\in\mathcal{I}}\Lambda_j(x_{ji}) \tag{4.44}$$

where

$$\Lambda_j(x_{ji}) = log\frac{\mathbb{P}(x_{ji}|\mathcal{H}_1)}{\mathbb{P}(x_{ji}|\mathcal{H}_0)}. \tag{4.45}$$

4.5.1 LRT for multiple subbands

In the case where we know b_{j1}, $\forall j \in \mathcal{I}_{sub}$, from the equations 4.15 and 4.44, we obtain:

$$\Lambda(X_{j,\forall j\in\mathcal{I}_{sub}}) = \mathcal{C} + \sum_{j\in\mathcal{I}_{sub}}\left(\frac{1}{b_{j0}} - \frac{1}{b_{j1}}\right)\sum_{i\in\mathcal{I}}x_{ji} \tag{4.46}$$

where $\mathcal{C}$ is a constant. For $j \in \mathcal{I}_{sub}$, $h \in \{0,1\}$, denote:

$$m_{jh} = Nab_{jh}\frac{b_{j1} - b_{j0}}{b_{j0}b_{j1}}, \tag{4.47}$$

$$v_{jh} = Nab_{jh}^2\frac{(b_{j1} - b_{j0})^2}{b_{j0}^2 b_{j1}^2}. \tag{4.48}$$

We have demonstrated in the previous section (ref. eq.4.16) that, under hypothesis $\mathcal{H}_k$, we have:

$$\left(\frac{1}{b_{j0}} - \frac{1}{b_{j1}}\right)\sum_{i\in\mathcal{I}} x_{ji} \sim \mathcal{N}(m_{jh}, v_{jh}). \tag{4.49}$$

Hence, under $\mathcal{H}_h$:

$$\Lambda(X_{j,\forall j\in\mathcal{I}_{sub}}) \sim \mathcal{N}(m_h, v_h). \tag{4.50}$$

where $m_h = \mathcal{C} + \sum_{j\in\mathcal{I}_{sub}} m_{jh}$, and $v_k = \sum_{j\in\mathcal{I}_{sub}} v_{jh}$. Let define $\Lambda^\star(X_{j,\forall j\in\mathcal{I}_{sub}}) = \frac{\Lambda(X_{j,\forall j\in\mathcal{I}_{sub}}) - m_0}{\sqrt{v_0}}$, the decision rule is now exactly the same as defined in the equation 4.20. The theorem 3 is also verified in this case.

4.5.2 GLRT for multiple subbands

In this case, b_{j1}, $j \in \mathcal{I}_{sub}$ are unknown. An estimation of b_{j1} can be obtained from the sample as follows:

$$\hat{b}_{j1} = \frac{1}{Na}\sum_{i\in\mathcal{I}} x_{ji} \tag{4.51}$$

By replacing b_{j1} by their consistent estimates $\hat{b}_{j1}$ in the equation 4.44, we obtain an estimate of $\Lambda(X_{j,\forall j\in\mathcal{I}_{sub}})$ as follows:

$$\hat{\Lambda}(X_{j,\forall j\in\mathcal{I}_{sub}}) = \mathcal{C} + \sum_{j\in\mathcal{I}_{sub}} \frac{N}{b_{j0}}\left[\bar{X}_j - ab_{j0}log(\bar{X}_j)\right] \tag{4.52}$$

where $\mathcal{C}$ is a constant which will be omitted in the next developments, and

$$\bar{X}_j = \frac{1}{N}\sum_{i\in\mathcal{I}} x_{ji}.$$

For $h \in \{0,1\}$, denote that:

$$\bar{X}_{jh} = \frac{\bar{X}_j - \mu_{jh}}{\sigma_{jh}} \tag{4.53}$$

where $\mu_{jh} = ab_{jh}$, and $\sigma_{jh} = \sqrt{\frac{a}{N} b_{jh}}$. Like in the equation 4.29, we obtain:

$$\bar{X}_{jh} \sim \mathcal{N}(0,1) \tag{4.54}$$

From the equations 4.52 and 4.53, by the same argument as given in the previous section to obtain the equation 4.32, we have:

$$\hat{\Lambda}(X_{j,\forall j \in \mathcal{I}_{sub}}) \approx \sum_{j \in \mathcal{I}_{sub}} \frac{1}{2} \left(\bar{X}_{jh} + c_{jh}\right)^2 + \sum_{j \in \mathcal{I}_{sub}} d_{jh}, \tag{4.55}$$

where c_{jh} and d_{jh} are constant defined as follows:

$$c_{jh} = \sqrt{Na} \frac{b_{jh} - b_{j0}}{b_{j0}}, \tag{4.56}$$

$$d_{jh} = Na \frac{b_{jh} - b_{j0} log(ab_{jh})}{b_{j0}} - \frac{1}{2} c_{jh}^2. \tag{4.57}$$

Note that, $c_{j0} = 0, \forall j \in \mathcal{I}_{sub}$. Let denote:

$$\hat{\Lambda}^\star(X_{j,\forall j \in \mathcal{I}_{sub}}) = 2 \left(\hat{\Lambda}(X_{j,\forall j \in \mathcal{I}_{sub}}) - \sum_{j \in \mathcal{I}_{sub}} d_{j0} \right) \tag{4.58}$$

then

$$\hat{\Lambda}^\star(X_{j,\forall j \in \mathcal{I}_{sub}}) \approx \sum_{j \in \mathcal{I}_{sub}} \bar{X}_{j0}^2. \tag{4.59}$$

It is easy to obtain the following theorem.

Theorem 6 *Under the hypothesis $\mathcal{H}_0$, denote N_{sub} the size of $\mathcal{I}_{sub}$, we have:*

$$\hat{\Lambda}^\star(X_{j,\forall j \in \mathcal{I}_{sub}}) \to \chi^2(N_{sub}) \tag{4.60}$$

Following Neyman-Pearson lemma, the decision rule is then defined as follows

$$\tilde{\delta}(X_{j,\forall j \in \mathcal{I}_{sub}}) = \begin{cases} \mathcal{H}_0 \text{ if } \hat{\Lambda}^\star(X_{j,\forall j \in \mathcal{I}_{sub}}) < \tilde{\tau} \\ \mathcal{H}_1 \text{ if } \hat{\Lambda}^\star(X_{j,\forall j \in \mathcal{I}_{sub}}) \geq \tilde{\tau} \end{cases} \tag{4.61}$$

where, in order to ensure that $\tilde{\delta}$ is in the class $\mathcal{K}_{\alpha_0}$, $\tilde{\tau}$ is the solution of the following equation:

$$\mathbb{P}_{\mathcal{H}_0} \left(\hat{\Lambda}^\star(X_{j,\forall j \in \mathcal{I}_{sub}}) \geq \tilde{\tau} \right) = \alpha_0. \tag{4.62}$$

Theorem 7 *The threshold decision $\tilde{\tau}$ and the power function $\beta_{\tilde{\delta}}$ of the test $\tilde{\delta}$ are given as follows:*

$$\tilde{\tau} = \Phi^{-1}_{\chi^2_{N_{sub}}}(1-\alpha_0), \tag{4.63}$$

$$\beta_{\hat{\delta}} = 1 - \Phi_{\chi^2_{N_{sub},\lambda}}\left(\tilde{\tau} + 2\sum_{j\in\mathcal{I}_{sub}}(d_{j0}-d_{j1})\right), \tag{4.64}$$

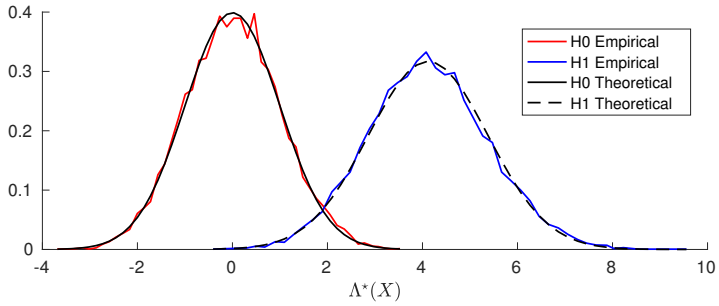
where $\Phi^{-1}_{\chi^2_{N_{sub}}}(.)$ denotes the inverse cumulative distribution function of a chi-square random variable with N_{sub} d.f; $\Phi_{\chi^2_{N_{sub},\lambda}}$ denotes the cumulative distribution function of a non-central chi-square random variable with N_{sub} d.f and the non-centrality parameter $\lambda = \sum_{j\in\mathcal{I}_{sub}} c^2_{j1}$.

4.6 Numerical Experiments

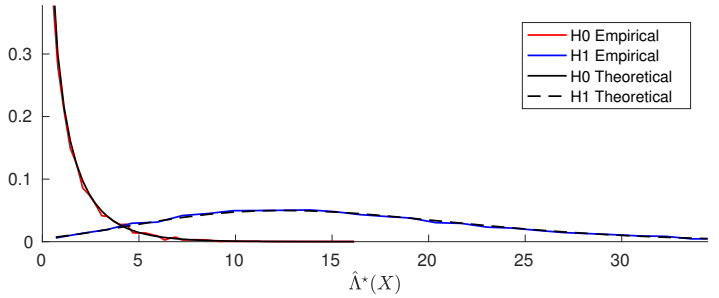
The construction of a real images database consumes too much time, especially in the case of our study, where we need a large number of images to validate the detectors proposed in the previous section. Therefore, we decided to test the proposed detectors first on a simulated database. The proposed detectors were first validated by simulated data which satisfy the given hypothesis, see 4.6.1. In the section 4.6.2, we tested the proposed detectors on simulated images. The section 4.6.3 gives the classification results of our detectors on a small real images database.

4.6.1 Detector Validation

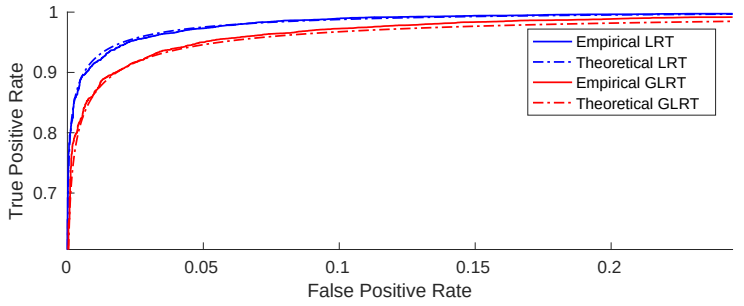
In order to make sure that the construction of our detectors is correct, we test these detectors on simulated data, where we know exactly all the parameters of the tests. We generate 20000 data samples, one half for $\mathcal{H}_0$ hypothesis and the other half for $\mathcal{H}_1$ hypothesis. Each data sample is composed by 64 Gamma random variable subsamples. These Gamma random variables are characterized by a shape constant of $1/2$, and by a scale constant b_{ji}, where $j \in \{1,..,64\}$ denotes the subsample index, and $i \in \{0,1\}$ denotes the hypothesis index. The set of $\{b_{ij}\}$ is estimated from simulated images used in the section 4.6.2. Without the loss of generality, we took the first subsample to build the OSSB detectors. All the 64 subsamples were taken to build the MSSB detectors. Figure 4.6 gives a comparison between the empirical and theoretical distributions of the statistics proposed in the OSSB detectors, and a



(a) Distribution of the stastistic proposed in the LRT



(b) Distribution of the stastistic proposed in the GLRT



(c) ROC curvres

Figure 4.6: (a),(b) - Empirical and theoritical distributions of the statistics proposed in the OSSB detectors , and (c) - the performance comparison between these detectors.

performance comparison between these detectors. Each empirical curve is well fitted by its theoretical counterpart. There is some loss of detection power observed in the case of the GLRT detector in comparison to the LRT detector. The lack of prior knownlegde about the $\mathcal{H}_1$ hypothesis is the main cause of this detection power loss. We also obtained the same results for the MSSB detectors. Theses detectors outperform their OSSB-version counterpart. The experimentations also showed that the performance of the proposed detectors is proportional to the subsample size.

4.6.2 Classification on simulated images

Print & Scan process introduces many kinds of distortions [53]. To produce simulated images, we have to mimic digitally the Print & Scan process, which is very complicated to model. There is no work in the literature which provides a complete model of the whole process. We separate the P&S process into two successive ones: the printing one and the acquisition one. The printing process involves many complicated digital and physical operations. To obtain a precise simulation of the process, it has to study each of these operations. We don't have the ambition to simulate the whole process because it is out of our expertise. In this study, by the limit of our knowledge, we tried only to produce some identified distortions in original images to obtain their simulated printed version. These operations are introduced in the Figure 4.7 and in the next paragraphs. The proposed simulation of the printing

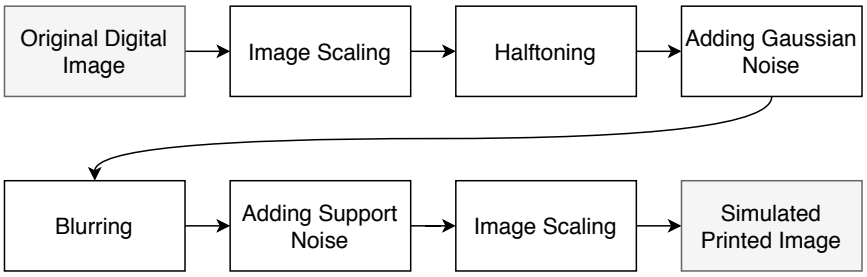


Figure 4.7: Proposed simulation of the printing process.

process starts by scaling the initial digital image. In fact, due to the different nature of images in the digital and physical spaces, the image must be resampled to be adapted with the printer's resolution. After that, because that printers, especially Laser or Ink-jet models, can only print out black dots, the image has to be processed by a reprographic technique (i.e., halftoning, dithering) before being physically processed. The reprographic operation permits to transforms a gray image into another presentation, which consists only of black patterns. Furthermore, other unidentified distortions may be introduced during the digital operations of the printing process, such as quantification errors, rounding errors, etc. In our simulation scheme, we assumed that these distortions can be modeled by a zero-mean Gaussian noise. There are also distortions introduced during the physical operations. For Ink-jet printing technique, ink drops are physical objects. When they are projected onto paper, they transform into 2D dots with bigger radius. There would be some superposition between nearby dots. We can use a blurring process (i.e., 2D median or Gaussian filter) to simulate this physical effect. Furthermore, printing paper is not uniform. Physical micro-textures of the paper will affect the appearance of the final printed image. In

our simulation, to take into account this deformation, some specific texture extracted from an image of a piece of real paper, is embedded into the simulated image. Real printed images are on physical support (i.e., paper). When we take a photo of a real object by a digital camera, photons emitted from the object would be projected onto the photovoltaic cells of the camera sensor. In function of the number of photons received, these cells will produce an electrical signal, which will be converted after that by a Analog/Digital Converter to produce a very first digital version of the image. Hence, this version is just a projection of the physical image onto a digital space. We assume that the camera sensor is ideally positioned so that the mentioned projection is orthogonal to the sensor plane. This projected version can be, therefore, considered as a scaling version of the printed one. The acquisition process is simply simulated by adding acquisition noise into images. Acquisition noises can be modeled by a heteroscedastic model proposed by Thai et al. in [115, 100]. The model provided for RAW images is employed for the sake of simplicity. This model is characterized by a couple of parameters (a, b), which models the linear relation between the variance and the expectation of image noise. The final simulated image is finally obtained by adding heteroscedastic noise on the image obtained from the previous operation.

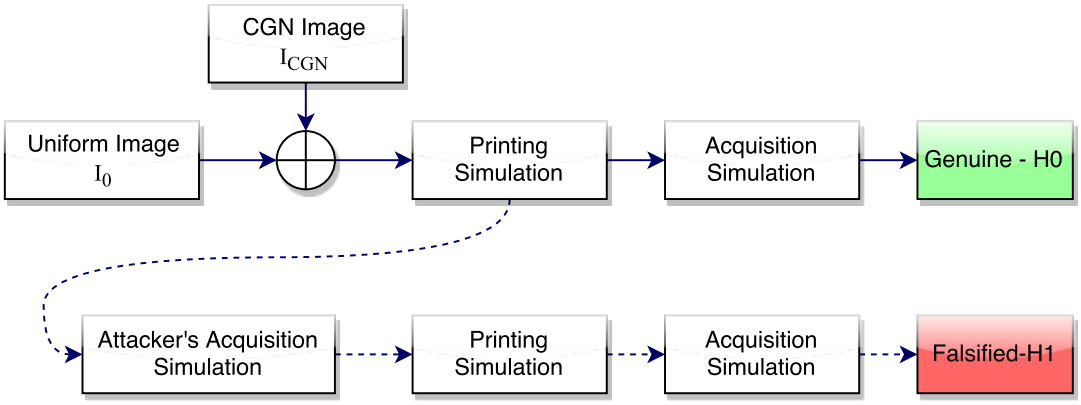


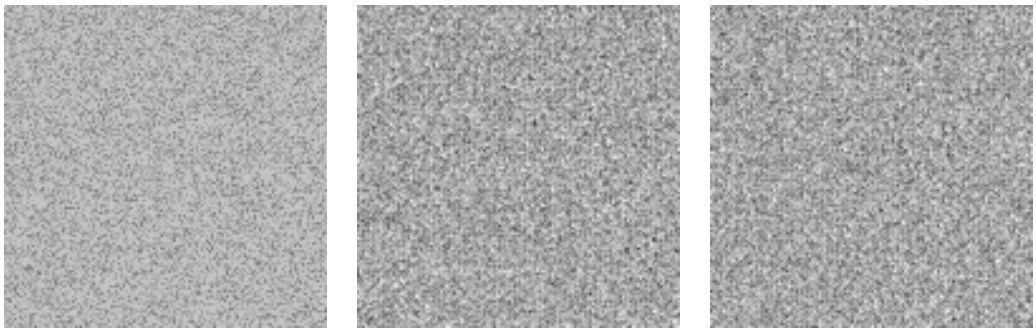
Figure 4.8: Construction flowchart for simulated images.

In our study, we have to create simulated version of genuine and falsified images. Figure 4.8 describes briefly the proposed construction flowchart of simulated images. Genuine images go through one P&S process. Meanwhile, falsified images go through two successive P&S process. In reality, the first scanning process and the second printing process are driven by counterfeiters to produce falsified codes from genuine codes. In general, these processes are not identical to the ones used in the creation of genuine codes. For the sake of simplicity, we did not integrate black modules of QR codes in our simulated images. Samples of simulated images are given in

the figure 4.9. We generated 2000 simulated images, half of them genuine, other half are falsified. The noise models of the main acquisition process and the counterfeiter's acquisition process are configured respectively by $(a_0, b_0) = (4.5e-5, 5e-7)$ and $(a_1, b_1) = (7.5e-5, 1e-6)$. For the printing simulation, numerical image is firstly scaled up by a factor of 2. Then, a dithering algorithm using uncluttered dots screen with 64 gray levels is applied. After adding some white noise, the image pass through a Gaussian filter to be blurred. Paper noise is extracted and embedded into the image after that. Finally, the image is scaled down by $\frac{1}{2}$ to obtain the final simulated printed image. The employed CGN texture is configured with the following parameters, see Table 4.1:

μ_0	ϵ	μ	σ^2
0.7	0.3	0.7843 or 200 $(uint8)$	0.3137 or 80 $(uint8)$

Table 4.1: Parameters of CGN textures



(a) Numerical **(b)** Genuine **(c)** Falsified

Figure 4.9: Sample of simulated images

The perceptual difference between simulated genuine and falsified images is hardly recognized. Figure 4.10 gives a comparison between the histogram of these two types of image. A light difference between these histogram is the reason that makes the related images look similar. The discrimination of simulated genuine and falsified images can be observed more clearly in the DCT domain. Figure 4.11 shows the distribution of DCT coefficients within the subband $(1, 2)$ under different hypothesis. Similar results are also obtained for the other DCT subbands.

Both OSSB and MSSB detectors were studied in this simulation. Without loss of generality, the subband $(1, 2)$ is opted to build the OSSB detectors. For the MSSB detectors, all the 64 subbands were used. In all of the cases, we obtained empirical distributions which were fitted correctly by their theoretical counterpart, Figure 4.12.

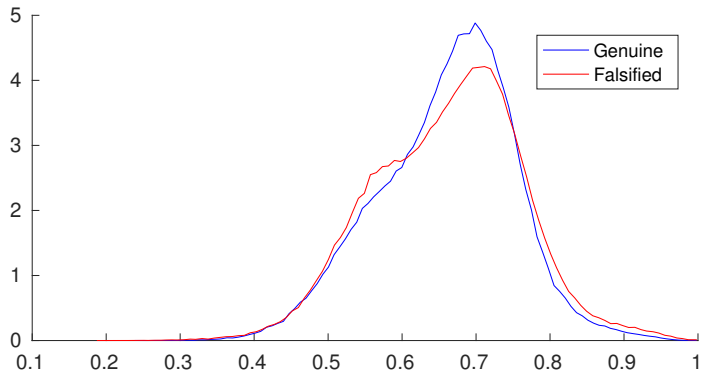


Figure 4.10: Histogram of samples of genuine and falsified simulated images.

The numerical results show that our proposed detectors have very high performance. The MSSB detectors behave better than the OSSB ones.

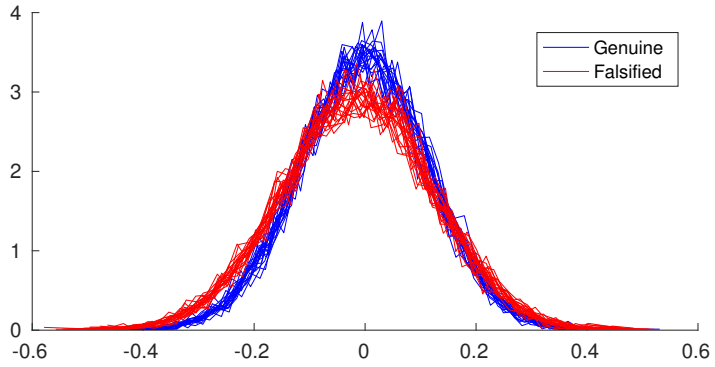


Figure 4.11: Distribution of DCT coefficients of images within subband $(1,2)$ under different hypothesis. Each curve corresponds to one image.

We also implemented an SVM classificator using the LBP descriptors as described in [46] to classify simulated images. This classificator is used to compare with our proposed detectors. The SVM-LBP detector need to have data from both of the two hypothesis for training. So that, to be fair, we compare the SVM-LBP detector with our MSSB-LRT one. Figure 4.13 gives the classification performance comparison between our proposed MSSB detectors and SVM-LBP one for simulated images of size 64×64 and 128×128. In both of the cases, the MSSB-LRT detector outperforms the SVM-LBP one. In the case of 128×128 image size, the MSSB-LRT detector manages to propose a perfect classification.

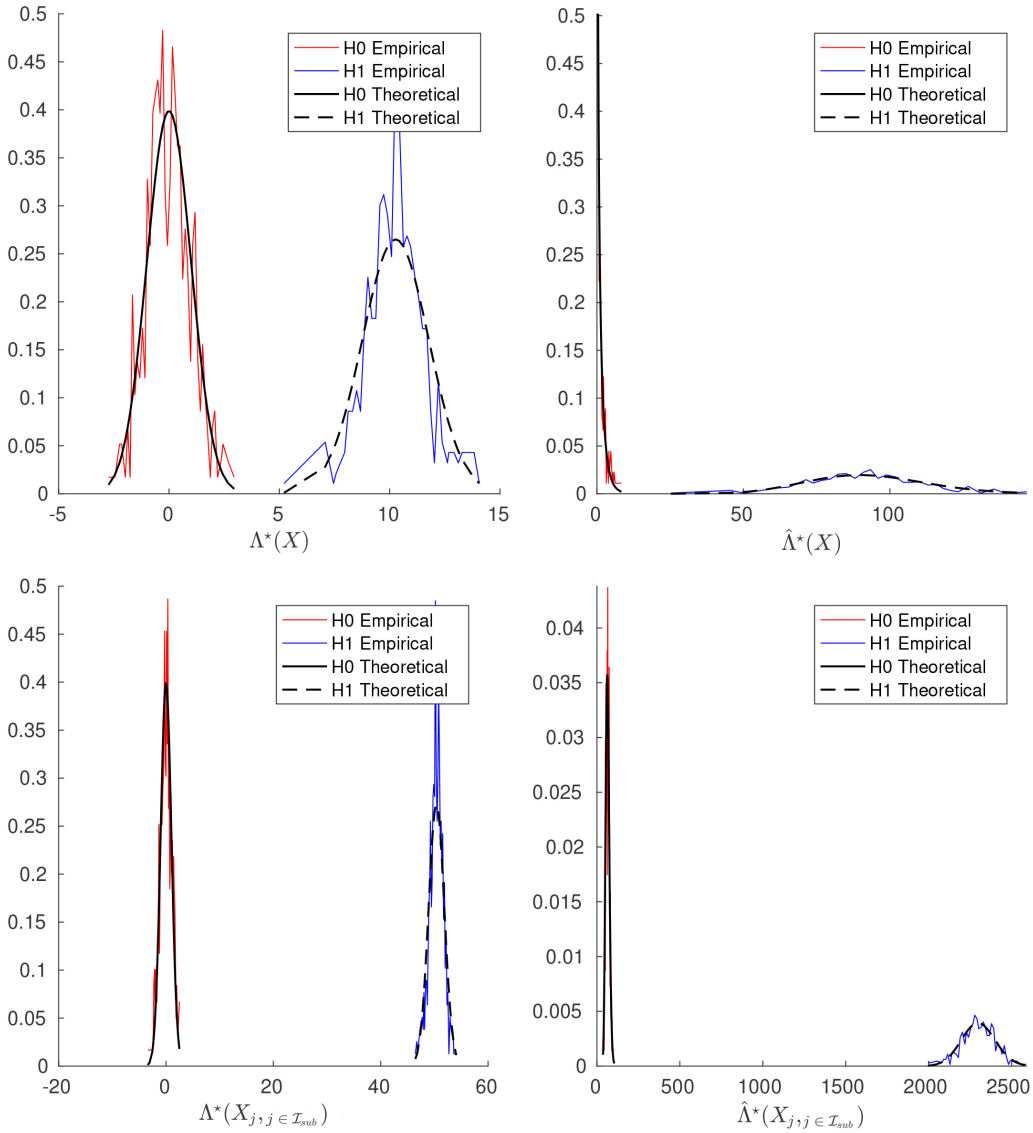
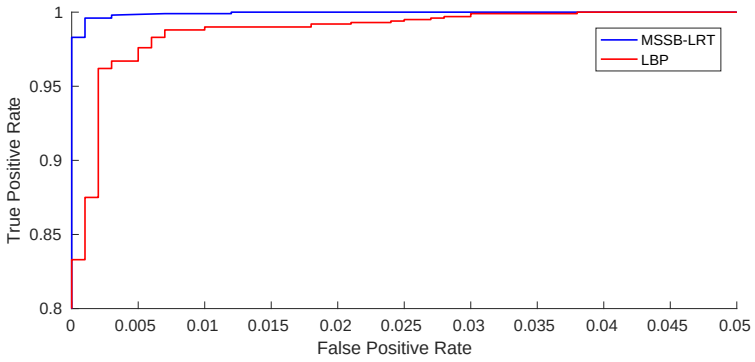


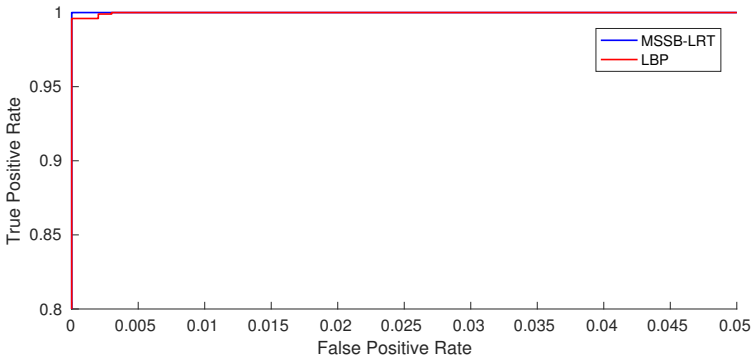
Figure 4.12: Empirical and theoretical distributions of the different proposed tests statistic introduced in the proposed detectors: OSSB-LRT (top-left), OSSB-GRLT (top-right), MSSB-LRT (bottom-left), and MSSB-GLRT (bottom-right).

4.6.3 Classification on real images

For real W-QR codes, only blocks situated in the background are interested. For the sake of simplicity, images without QR codes are used in this experiment. By using the same configuration as used in the simulation case, we created 200 digital images. These images were then printed out by a laser printer in 600 dpi on standard A4 papers to form 200 original 3x3-cm printed samples. After that, by scanning and reprinting each original printed sample, we obtained 200 falsified printed samples. All the original and falsified printed samples were then digitalized by a Iphone 7 camera at its highest resolution under an ISO constant. The distance between the



(a) Image size 64×64



(b) Image size 128×128

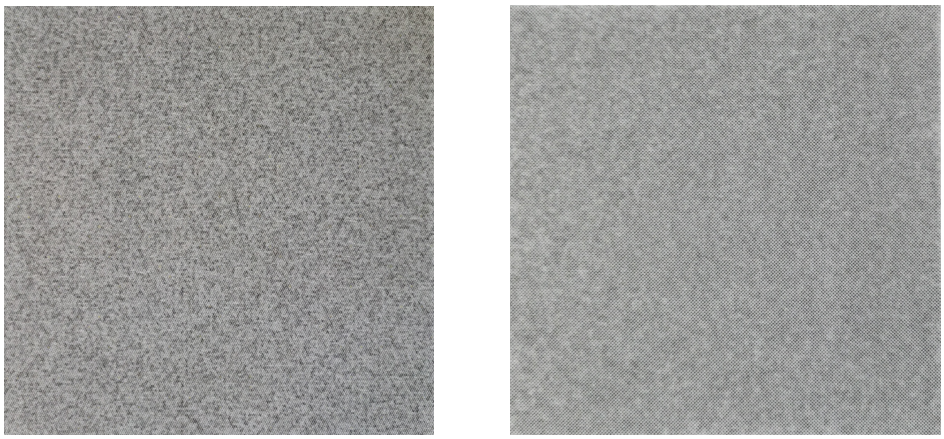
Figure 4.13: Classification performance comparison between MSSB detectors and SVM-LBP one for different sizes of simulated images

camera and the printed sample is maintained at a constant value. All the images are taken under the best focus conditions. They are cropped to have the size 512×512. Figure 4.14 shows a sample of genuine and falsified images.

We took the $(1,2)$ DCT channel to build the OSSB detectors and all the 64 DCT channels to build the MSSB detectors. Figure 4.15 gives the empirical ROC curves of the proposed detectors. We manage to classify all images using the MSSB detectors correctly.

4.7 Conclusion

In this chapter, to use QR codes as an anti-counterfeiting solution for product labeling, we introduced a watermarking technique to secure them. A security level, which can prevent the illegal cloning of standard QR codes, is added by substituting the white background of the codes by a specific texture. The printing process will affect the visual behavior of the embedded texture by irreversibly degrading it.



(a) Genuine **(b)** Falsified

Figure 4.14: Sample of real images

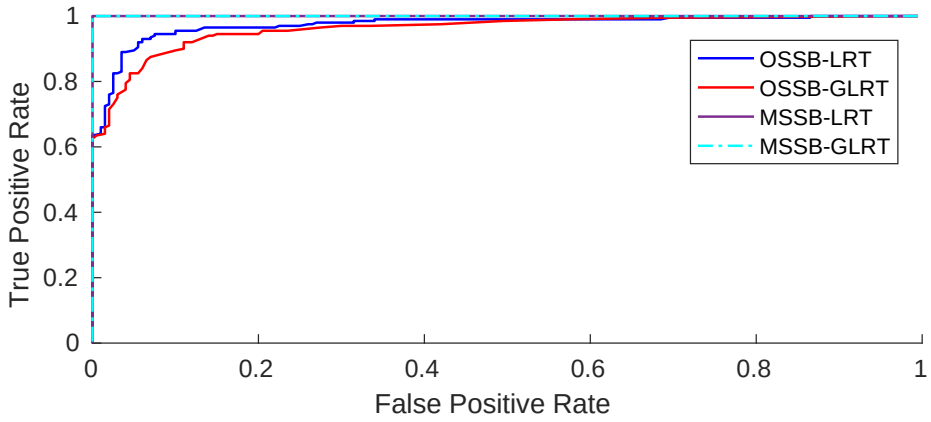


Figure 4.15: Empirical performance of the proposed detectors on real images database.

Counterfeiters need to have the digital codes and the same printing system as the constructor to produce valid secured codes, which is highly sophisticated. Furthermore, cloning an existing code by scanning and reprinting will change the behavior of embedded texture, and can be detected. Then, by analyzing the embedded texture, we can verify the validity of the printed QR codes, and by consequence, the genuity of the related product. We had the idea for the proposed solution when we worked on face PAD problems, described in the Chapter 3. The idea is about creating some sort of textural image which is fragile under the P&S process so that any attempt to duplicate the image by scanning and reprinting will degrade it. This degradation was supposed to be detectable with the Noise Local Variance proposed in the previous chapter. We discovered finally the CGN texture. However, we also discovered some limitations of the Noise Local Variance model related to its required assumptions, which make it less effective when applied in the authentication of CGN texture problem. The Subsample Variance Model, a generalized version of Noise Local Variance

model, is then proposed. The new model works well for sample of any identically and independently distributed random variable. We have applied the proposed model to characterize the DCT coefficient distribution, which helps us after that building powerful statistical detectors using the framework of statistical hypothesis testing.

Chapter 5

Image forgery detection solution based on JPEG compression signature in DCT domain

Contents

5.1 Introduction

JPEG is a widely used image format which is appropriate for storage and transmission purposes. Distortions caused by JPEG compression can help forgers to better hide their manipulations by disrupting several useful image regularities, such as noise, aberrations, etc. Therefore, the workflow of a digital image forgery often finishes by intentionally or unintentionally resaving the forged image under the JPEG

format. If the original image is already in JPEG format, the final forged image would contain regions where a double compression has been occurred. JPEG compression introduces some particular properties systematically in the DCT domain. The section 5.2 gives a description of these properties under different situations where a single or a double JPEG compression has been occurred. We introduce after that, in the section 5.3 a histogram-based solution for the estimation of the first quantization step employed in a double-compressed image. In the section 5.4, we introduce a new and simple solution for image forgery detection based on the estimation of the first quantization step. The proposed solution has been tested on simulated data and images from some public databases. Experimental results show a good performance of the proposed solution for detecting forged images compressed under high quality factor.

5.2 Signature of JPEG compression in the DCT domain

5.2.1 JPEG compression

After optionally applying the color space conversion and downsampling, JPEG compression process continues by dividing each color channels into non-overlapped 8×8 blocks. Each block is first transformed in DCT domain. The DCT coefficients of each block will be converted into entire numbers and then quantized. For a given 8×8 blocks of non-compressed image, let denote $m_i, i \in \{1,2,..64\}$ the i-th coefficient obtained after applying DCT on the block. Each coefficient is related to the contribution of an associated frequency in the block. The quantization operation is performed to reduce information in high frequency components, which are often less sensitive to the human visual system. Let denote $q_i, i \in \{1,2,..64\}$ the quantization steps employed for quantizing the 64 corresponding DCT coefficients. The value of q_i is constant over all the blocks. It is defined *a priori* in function of the expected Quality Factor. The Quantization Table, constituted by $q_i, i \in \{1,2,..64\}$, is stocked as metadata of the JPEG file, which can be easily retrieved during the decompression process. For a given DCT coefficient, by omitting its index for the sake of simplicity, denote m, c, Q respectively the DCT coefficient, its quantized value and the corresponding quantization step. We have that:

$$c = \left[\frac{m}{Q}\right]. \tag{5.1}$$

The JPEG decompression is processed by applying dequantization operation and then the DCT inverse transform to come back in spatial domain. The dequantized DCT coefficient, denoted d, is obtained by multiplying its corresponding quantized value with the same quantization step used during compression process. We have that:

$$d = cQ = c\left[\frac{m}{Q}\right]. \tag{5.2}$$

Therefore, the dequantized DCT coefficients are multiples of the corresponding quantization steps.

5.2.2 JPEG double compression

The JPEG double compression involves two successive JPEG compression processes. Figure 5.1 illustrates different operations occurring in a JPEG double compression. White rectangles represent the different input/output images. Knowing that JPEG compression is processed independently between different 8×8 non-overlapped blocks, we use color rectangles to represent different stages of a block: blue for the first compression and red for the second one. Dashed lines designate the passage between blocks and images. The color space conversion requires blocks from all color channels to process. For a given DCT frequency, denote $Q_i, c_i, d_i, i \in \{1,2\}$ respectively

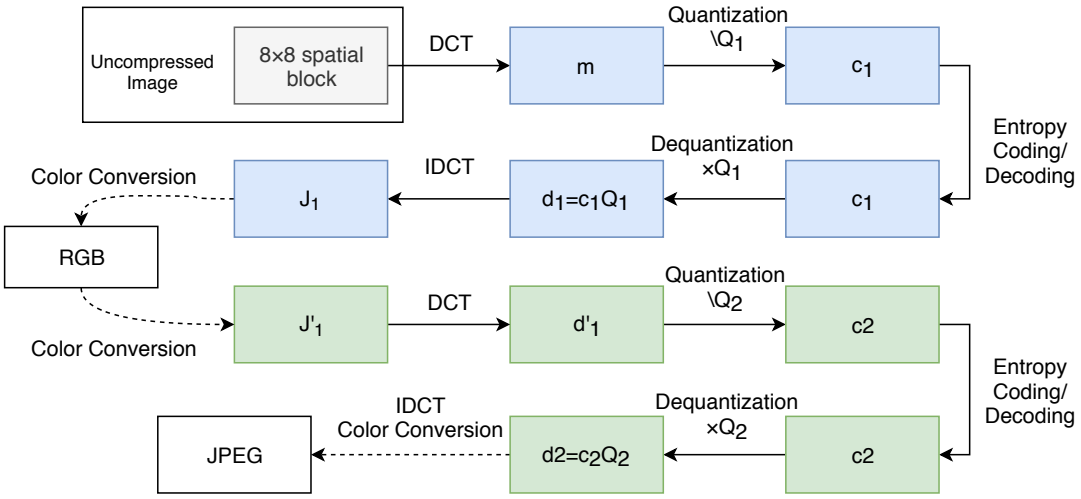


Figure 5.1: Different operations occurring in a JPEG double compression process. The index of DCT channels is omitted for the sake of simplicity.

the quantization step, the quantized and dequantized DCT coefficient of the i-th compression. Denote d_1' the DCT coefficient obtained after the second DCT transform. Denote respectively J_1, J_1' the 8×8 spatial blocks obtained right after the first IDCT transform and before the second DCT transform. Due to rounding error of the color

space conversion operations, J_1' may differ from J_1. We have that:

$$J' = J + \mathfrak{R}_{CSC}, \tag{5.3}$$

where $\mathfrak{R}_{CSC}$ is the cumulative errors of the two color space conversion processes. Noting that DCT and IDCT are linear operations, we obtain the following development:

$$\begin{aligned} d_1' &= \mathrm{DCT}(J_1') = \mathrm{DCT}(J_1 + \mathfrak{R}_{CSC}) \\ &= \mathrm{DCT}([\mathrm{IDCT}(d_1)] + \mathfrak{R}_{CSC}) \\ &= \mathrm{DCT}(\mathrm{IDCT}(d_1) + \mathfrak{R} + \mathfrak{R}_{CSC}) \\ &= d_1 + \mathfrak{E} = c_1 Q_1 + \mathfrak{E}. \end{aligned} \tag{5.4}$$

where $\mathfrak{E} = \mathrm{DCT}(\mathfrak{R} + \mathfrak{R}_{CSC})$, $\mathfrak{R}$ is the rounding error of the IDCT transform, and $[.]$ is the round-off function. $\mathfrak{E}$ can be considered as a random variable centered at 0, with a high probability of dropping in the $(-1,1)$ interval ($> 90\%$), [136]. Finally, we obtain the following equation of c_2:

$$c_2 = \left[\frac{c_1 Q_1 + \mathfrak{E}}{Q_2}\right]. \tag{5.5}$$

5.3 Quantization step estimation

Given a JPEG image, we can retrieve the quantized DCT coefficients easily and the Quantization Table from JPEG file. However, in double-compressed images, the quantization step and the exact values of quantized DCT coefficients involved in the first compression would be lost after the second one. For forensics purposes, it is proposed to estimate the lost Q_1 knowing Q_2 and having a sample of c_2. Lou et al. [137], Thai et al. [136] have proposed different solutions for estimating quantization step from an image that has been previously JPEG-compressed and then stored in lossless format. These solutions solve only a particular case of the proposed problem, when $Q_2 = 1$. Pevny and Fridirich [138] have proposed a method for the detection of double JPEG compression using a soft-margin support vector machine. The 144-dimensional features vector employed contains the number of occurrences of the first 16 multiples of the second quantization step retrieved from the first 9 AC frequencies. They have also proposed a multiclassifier which permits detecting Q_1 for the first 9 AC frequencies when $Q_2 \in \{4,5,6,7,8\}$. In this section, we propose a new

histogram-based approach for the estimation of Q_1 knowing Q_2 and having a sample of c_2. Let consider the Equation 5.5, knowing that $\mathfrak{E}$ is a random variable which drops into the $]-1,1[$ interval with a very high probability, we have that:

$$\left[\frac{c_1 Q_1}{Q_2}\right] - 1 \leq c_2 \leq \left[\frac{c_1 Q_1}{Q_2}\right] + 1. \tag{5.6}$$

and it is more likely that $c_2 = \left[\frac{c_1 Q_1}{Q_2}\right]$. It means that the histogram of a sample of c_2 will highly likely contains peaks located at $\left[\frac{kQ_1}{Q_2}\right], k \in \mathbb{Z}$. Furthermore, basing on the statistical model of DCT coefficients proposed by Thai et al [139], it is likely that:

$$h\left(\left[\frac{k_1 Q_1}{Q_2}\right]\right) > h\left(\left[\frac{k_2 Q_1}{Q_2}\right]\right) \tag{5.7}$$

for $0 \leq k_1 < k_2$ or $0 \geq k_1 > k_2$ where $h(x)$ denotes the number of occurrences of x in the histogram. Given Q_1, by varying k, we can predict all the possible values of $\left[\frac{kQ_1}{Q_2}\right]$, and then all the possible peaks of the histogram. Denote $\mathbf{P}_{Q_1} = \{p_i, i \in \mathbf{I}_{P_{Q_1}}\}$, and $\mathbf{H} = \{x_i, i \in \mathbf{I}_H\}$ respectively the sets of predicted peaks' location and actual peaks' location of $c2$ sample's histogram, where $\mathbf{I}_{P_{Q_1}}, \mathbf{I}_H$ are respectively the sets of index of $\mathbf{P}_{Q_1}$ and $\mathbf{H}$, we propose a measure of the difference between the set $\mathbf{P}_{Q_1}$ and the histogram of c_2 sample, denoted by $S(\mathbf{P}_{Q_1}, \mathbf{H})$, as follows:

$$S(\mathbf{P}_{Q_1}, \mathbf{H}) = \sum_{p \in \mathbf{P}_{Q_1}} (f_H(p) - h(p)) + \sum_{p \in \mathbf{H}} (1 - e_{P_{Q_1}}(p)) f_H(p) \tag{5.8}$$

where

$$e_X(p) = \begin{cases} 1 : \text{if } p \in \mathbf{X} \\ 0 : \text{if } p \notin \mathbf{X} \end{cases} \tag{5.9}$$

and $f_H(x)$ is the score function, which is empirically defined by the histogram of the $c2$ sample as follows:

$$f_H(\epsilon x_i + (1-\epsilon)x_{i+1}) = \epsilon h(x_i) + (1-\epsilon)h(x_{i+1}) \tag{5.10}$$

for all $\epsilon \in [0,1]$ and x_i, x_{i+1} are any two successive elements of $\mathbf{H}$. In the equation 5.8, the first-term accounts for the difference contributed by DCT values belonging to $\mathbf{P}_{Q_1}$, the second term accounts for the difference contributed by DCT values belonging to $\mathbf{H}$ but not belonging to $\mathbf{P}_{Q_1}$. DCT values which do not belong to $\mathbf{H} \cup \mathbf{P}_{Q_1}$ are not taken into account in the proposed measure of difference. Figure 5.2 shows a typical histogram of DCT coefficients (for a given frequency) knowing that $Q_1 = 15$

and $Q_2 = 2$. The score function $f_H(x)$ is given in red. Figure 5.3 illustrates different terms contributing to the measure of difference $S(\mathbf{P}_{Q_1}, \mathbf{H})$ where $Q_1 = 15$, $Q_2 = 2$ and the predicted value of Q_1 is set at 9. The value of Q_1 which minimizes the mea-

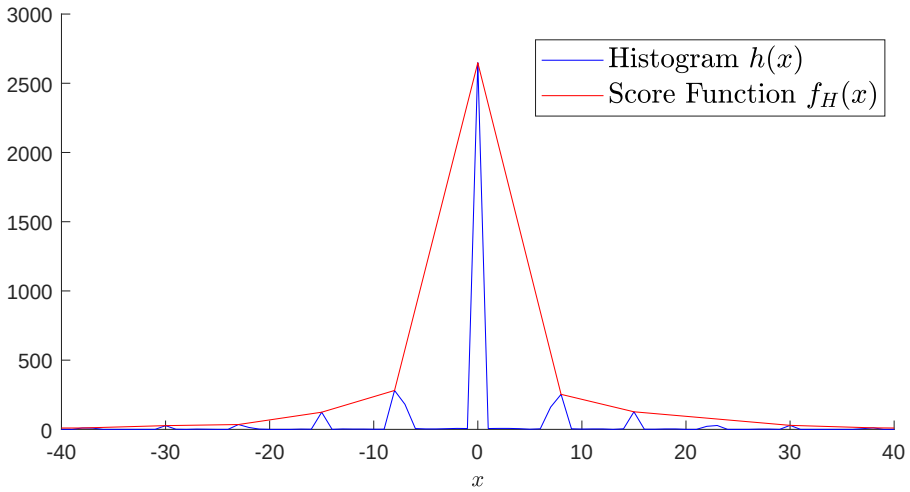


Figure 5.2: Histogram of DCT coefficients and the proposed scoring function ($Q_1 = 15$, $Q_2 = 2$)

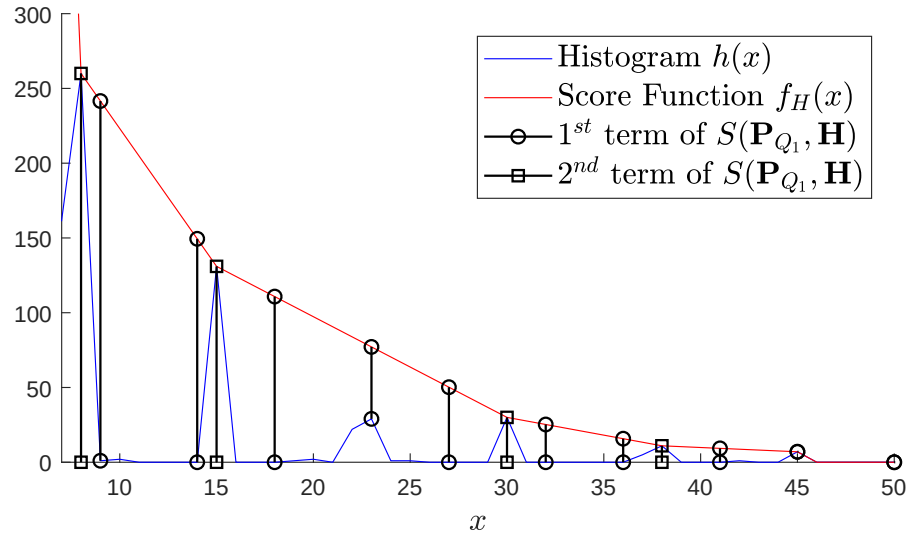


Figure 5.3: Illustration of different terms contributing in $S(\mathbf{P}_{Q_1}, \mathbf{H})$ where $Q_1 = 15$, $Q_2 = 2$ and the predicted value of Q_1 is set at 9.

sure of difference $S(\mathbf{P}_{Q_1}, \mathbf{H})$ is the best estimation for the unknown Q_1. The Figure 5.4 illustrates the evolution of the measure of difference $S(\mathbf{P}_{Q_1}, \mathbf{H})$ when Q_1 varies from 1 to 100. The proposed method for estimation of the first quantization step performs correctly when $Q_1 > Q_2$, which means that the Quality Factor (QF) of the second compression must be greater than the QF of the first one. Furthermore, the estimation of Q_1 in high DCT frequencies is not reliable due to insufficient statistics.

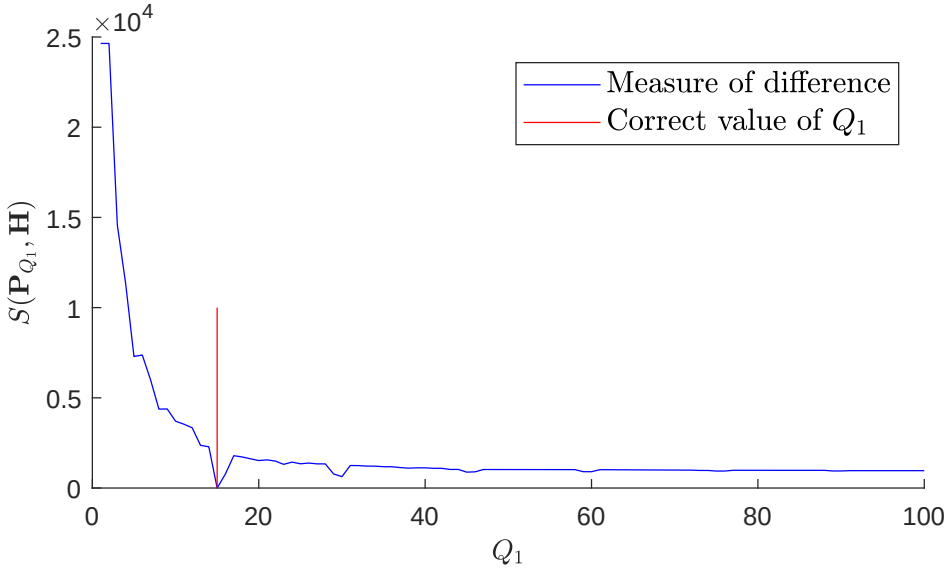


Figure 5.4: Typical evolution of the measure of difference $S(\mathbf{P}_{Q_1}, \mathbf{H})$ ($Q_1 = 15$, $Q_2 = 2$).

5.4 Proposed forgery detection solution

In this study, we only consider the cases where forged images are created by modifying locally an authentic JPEG image, titled here as the carrier image. When a region of the carrier image is forged, because of many reasons, the DCT coefficients within the forged region do not have the same behavior of the ones in the rest of the image. The reasons could be:

- The forged region comes from an uncompressed image or an image compressed using Quantification Table different from the one utilized in the carrier image.
- Manipulations such as filtering, interpolating, scaling, etc. during the forgery process break the characteristics of quantized DCT coefficients of the forged region.
- There may be some mismatch of the DCT grid of the forged region with that of the rest of the image. When the forged image is recompressed to save, the forged region is forced to use the same DCT grid as the whole image. Traces of the first compression will disappear for blocks within forged region.

Figure 5.5 shows the normalized histograms of DCT coefficients at a given frequency of a forged image and of the forged region within the same image. When the area of the forged region is small enough in comparison to the whole image's area, the

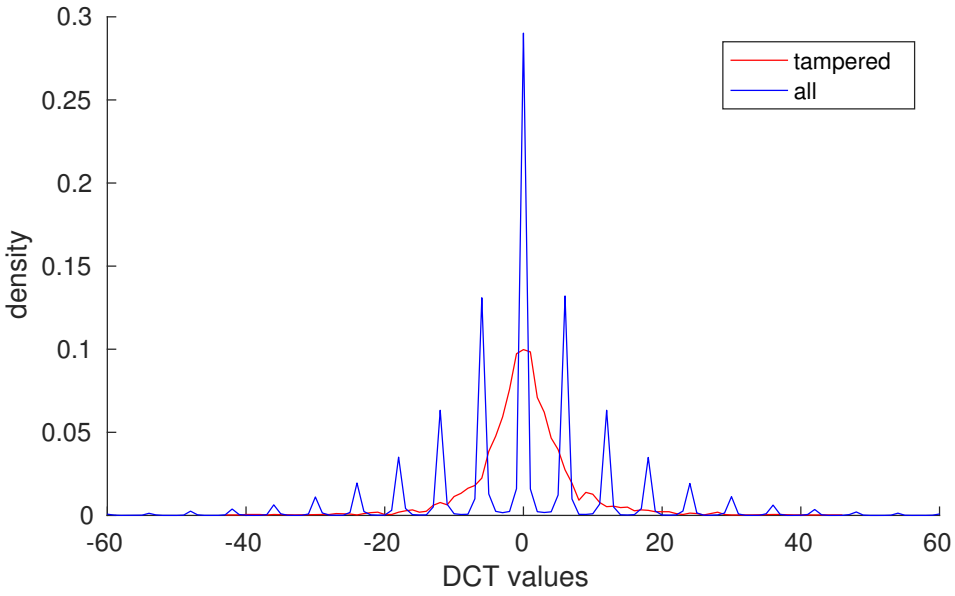


Figure 5.5: Normalized histograms of DCT coefficients at a given frequency of a forged image (blue curve) and of the forged region within the same image (red curve).

DCT coefficients of blocks within the forged region cannot contribute to changing the shape of the whole image's DCT histograms. Then, we can employ these DCT histograms to estimate the quantization steps of the JPEG compression realized previously on the carrier image. For a given DCT channel, having the corresponding estimated quantization step, we can predict all the values possible of its DCT coefficients, $\mathbf{P} = \{p_i, i \in \mathbf{I}_P\}$. If the estimation is precise, blocks of which the corresponding DCT coefficient does not belong to the set of predicted DCT values can be considered as forged ones. Denote $\mathbf{Z} = \{z_i | i \in \mathbf{I}_B\}$ the set of all DCT values belonging to the channel, where I_B is the set of 8×8 block indexes. We propose a score map $\mathbf{S} = \{s_i | i \in \mathbf{I}_B\}$ for forgery detection, which is defined as follows:

$$s_i = f_P(z_i). \tag{5.11}$$

where the score function $f_S(z)$ is defined as follows:

$$f_P(z) = \begin{cases} d_P(z) & \text{if } d_P(z) > 1 \\ 0 & \text{if } d_P(z) \in \{0, 1\} \end{cases} \tag{5.12}$$

and the function $d_P(z)$ returns the distance from z to the nearest element of $\mathbf{P}$. For any z, there always exists p and q $(p < q)$ two successive elements of $\mathbf{P}$ and $\alpha \in$

$[0,1)$ such as $z = \alpha p + (1-\alpha)q$, $d_P(z)$ is defined as follows:

$$d_P(z) = min(\alpha, 1-\alpha)(q-p). \tag{5.13}$$

The score function $f_P(z)$ is set to the distance $d_P(z)$, because we simply consider that the more a DCT value is far from its nearest $p \in \mathbf{P}$, the more likely that it belongs to the forged region. When $d_P(z) \leq 1$, it is highly likely that the related block belongs to unforged regions. To minimize the false detection rate, $f_P(z)$ is set to zero in these cases. Hence, for every DCT channels where we can estimate the first

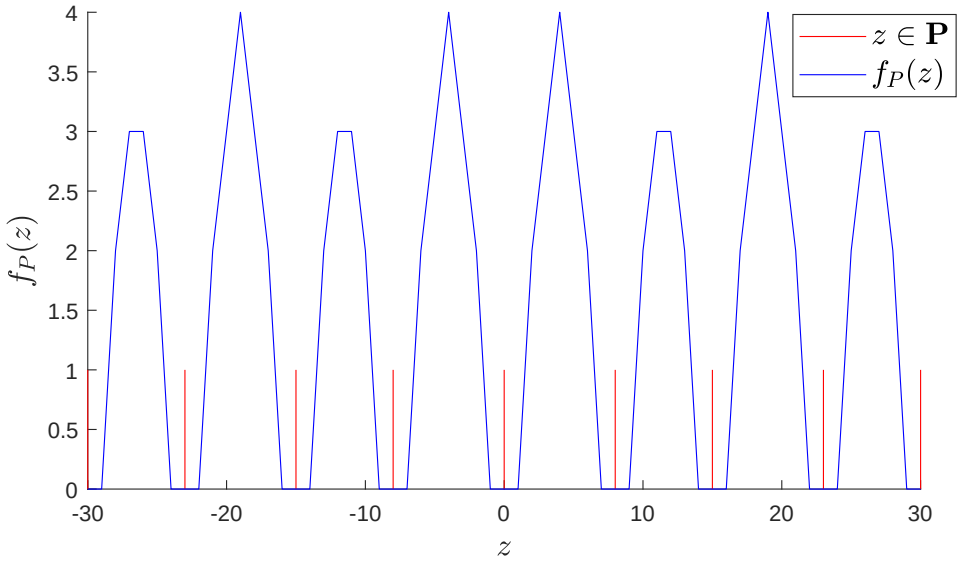


Figure 5.6: Illustration of the function $f_P(z)$ where the set $\mathbb{P}$ is defined with $Q_1 = 15$ and $Q_2 = 2$.

quantization step, we can produce a score map. By summing up all these individual score maps, we obtain a global score map, which permits detecting and localizing forgeries. Figure 5.11 shows a set of individual score maps and the combined global one of a forged image. The ground truth of the given image is also provided for comparison. The presence of clusters of non-zero scores in the global score map permits highlighting and localizing forged regions. However, a zero score map cannot provide us sufficient means to confirm the authenticity of an image.

5.5 Numerical Results

5.5.1 Estimation of the first quantization step

To evaluate the performance of the proposed solution for the estimation of the first quantization step, for each couple of (Q_1, Q_2) $Q_1 \in \{1, 2, ..., 100\}$ and $Q_2 \in \{1, 2, ..., 50\}$, we generate a sample of c_2. The generated sample of c_2 would be used to obtain an estimated value of Q_1, denoted $\hat{Q}_1$. We finally compare $\hat{Q}_1$ and its correct counterpart Q_1. The sample of c_2 contains 4096 elements, which corresponds to the number of 8×8 blocks of a 512×512 image. Each element of the sample is obtained by the following equation:

$$c_2 = \left\lfloor \frac{\left\lfloor \frac{m}{Q_1} \right\rceil Q_1 + \mathfrak{E}}{Q_2} \right\rfloor, \tag{5.14}$$

where m represents a value of a non-quantized DCT coefficient, which is modeled as a sample of a zero-mean Laplacian random variable with the scale parameter $b = 128$, knowing that the probability density function of a zero-mean Laplacian random variable with the scale parameter b is given as follows:

$$f(x) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right), \text{ where } x \in \mathbb{R}, \tag{5.15}$$

and $\mathfrak{E}$ is a zero-mean random variable where

$$P(\mathfrak{E} \in (-1, 1)) = 0.9.$$

Figure 5.7 shows the values of $|\hat{Q}_1 - Q_1|$ for different values of Q_1 and Q_2. The proposed algorithm can define precisely in most of the cases when $Q_1 > 2Q_2$. When $Q_2 < 15$, the proposed algorithm can also define precisely some value of Q_1 which is smaller than $2Q_2$ and greater than Q_2. When $Q_1 = 2Q_2$ or $Q_1 < Q_2$, the proposed algorithm fails to estimate Q_1. The bigger the value of Q_2 is, the more the performance of the proposed estimation algorithm decreases.

The proposed estimation method relies on the dynamic in value of the c_2 sample. If the distribution of m has a small scale parameter and when Q_1 and Q_2 are big enough, all or most of the value of c_2 sample will be zero. The other values of c_2, if exist will create one or two small peaks far from the principal peak, which is located at 0. The poor dynamic of the related histogram does not permit to obtain a good

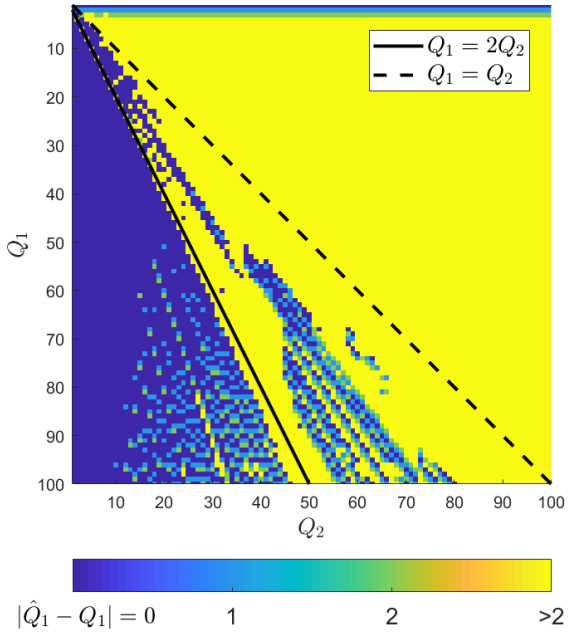


Figure 5.7: Performance of the proposed method for the estimation of Q_1 for different values of Q_2 visualized via the value of $|\hat{Q}_1 - Q_1|$.

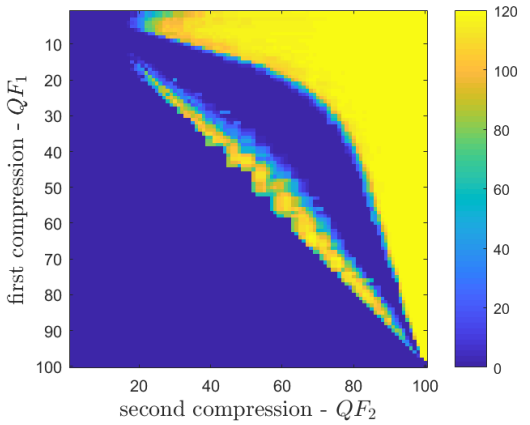
estimation using the proposed method. The size of the c_2 sample is also a factor which impacts directly on the performance of the proposed method.

5.5.2 Forgery detection performance on simulated images

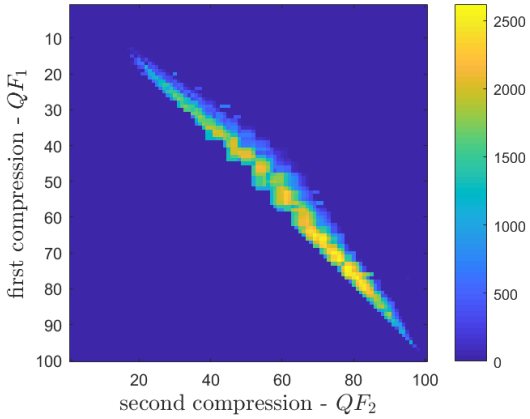
The proposed forgery detection solution has been first tested on simulated images. We have employed two 512×512 TIFF images, *boats512x512.tif* and *bridges512x512.tif*, for the simulation, Figures 5.8a and 5.8b. The *boats* and *bridges* images are respectively used as carrier and donner for a splicing operation. The two images are respectively compressed with a quality factor equal to QF_1 and 95. Assuming that the forger possesses only the compressed version of the two images. The forged image, Figure 5.8c, is simply created by copying a small square of the *bridge's* image and then pasting it on an arbitrary position within the *boat's* image. Figure 5.8d gives the ground-truth of the given forged image. After the splicing operation, the forger would resave the forged image under the JPEG format, and the QF selected for the JPEG compression is QF_2. Quality Factor of a JPEG compression varies from 1 to 100. When $QF = 1$, all the 64 quantization steps will be set to 255, the compressed image lost almost its information. When $QF = 100$, all the 64 quantization steps will be set to 1, there is no compression and the image is in its best quality. For

Figure 5.8: Simulation of splicing forgery: (a) Carrier image *boats512x512.tif*, (b) Donner image *bridges512x512.tif*, (c) forged image created by splicing, (d) ground truth.

different values of QF_1 and QF_2 varying from 1 to 100, we created a forged image. The proposed method for image forgery detection is then applied to the image to detect the forged region. We evaluated the detection performance by studying the number of forged block correctly detected and the number of unforged blocks incorrectly classified for different values of QF_1 and QF_2. Figure 5.9a and Figure 5.9b describe respectively the number of forged block correctly detected and the number of unforged blocks incorrectly classified for different values of QF_1 and QF_2. By comparing the two figures, we can see that the proposed solution performs well over a big set of a couple of (QF_1, QF_2). For every couple of (QF_1, QF_2) which locates in the yellow region in the top-right corner of Figure 5.9a, the detection of forged region is precise. For couples which locates on the hot color regions which present



(a)



(b)

Figure 5.9: Number forged blocks correctly detected (a) and number of unforged blocks incorrectly classified (b) for different values of Quality Factor of the first and second JPEG compressions. Simulated images contain 4096 8×8-blocks in total and the number of forged blocks is 120.

on both Figure 5.9a and Figure 5.9b, the proposed solution performs badly with a very high number of unforged blocks misclassified. The blue region represents all the couples (QF_1, QF_2) where the forged image can bypass the proposed detection solution.

5.5.3 Forgery detection performance on images from public database

As demonstrated in the previous section, we admit that our method cannot detect all forgeries. It can only detect forgery for JPEG images which were created from JPEG images and resaved with a QF greater than the QF of the original ones. When forged images is resaved under an uncompressed format such as TIFF, we can consider that $Q_2 = 1$ for all the 64 DCT coefficients. We can then obtain an estimated sample of

c_2 by applying the DCT manually transform. The proposed solution performs well in these cases. We have tested our image forgery detection solution on the two public databases CASIA v1 and CASA v2 [140]. For $QF_2 > 95$, our method performs well for most of the case, we restricted therefore to study images which have QF greater than 95. From the CASIA databases, we have extracted a set of 177 forged images. The proposed solution achieved to localize partially or entirely the forged region in 169 over 177 extracted image. Figure 5.12 shows examples where we can detect entirely the forged region of the image. The proposed solution is compared with Lin et al.'s solution [113]. The proposed solution gives a detection map much less noisy than the one given by [113]. Figure 5.10 shows examples where we can detect only a portion of forged region from the image. Most of undetected forged blocks are situated in homogeneous or saturated regions, where there is not too much information, and they cannot be exposed by the proposed solution, Figure 5.10a. For some images, forged and unforged regions are previously compressed under the same conditions, and the forged region's DCT grid matches correctly the one of the rest of the image. Blocks within the forged region of these images are undetected. However, forged the region in general is not created as a set of 8×8-blocks but as a cluster of pixels. Therefore, blocks situated in the boundaries of forged region can contain some pixels from the unforged one. This spatial combination changes obviously the characteristics of DCT coefficients of these blocks, which makes them exposed under the proposed detection method. When blocks on the boundaries region are detected, the forged region can also be detected and localized, see Figure 5.10b for example.

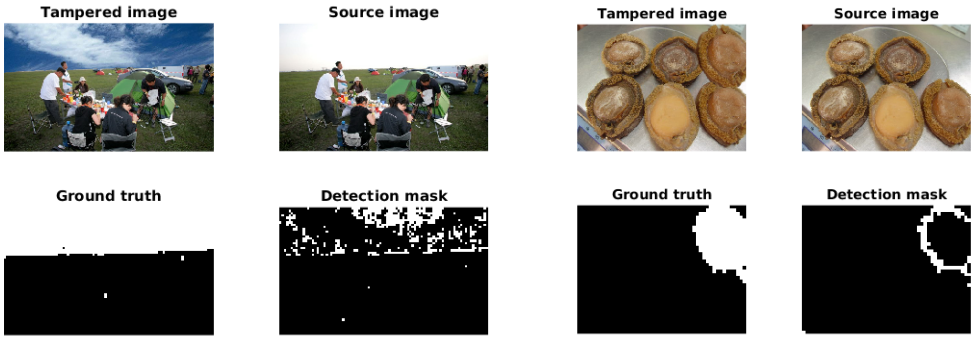
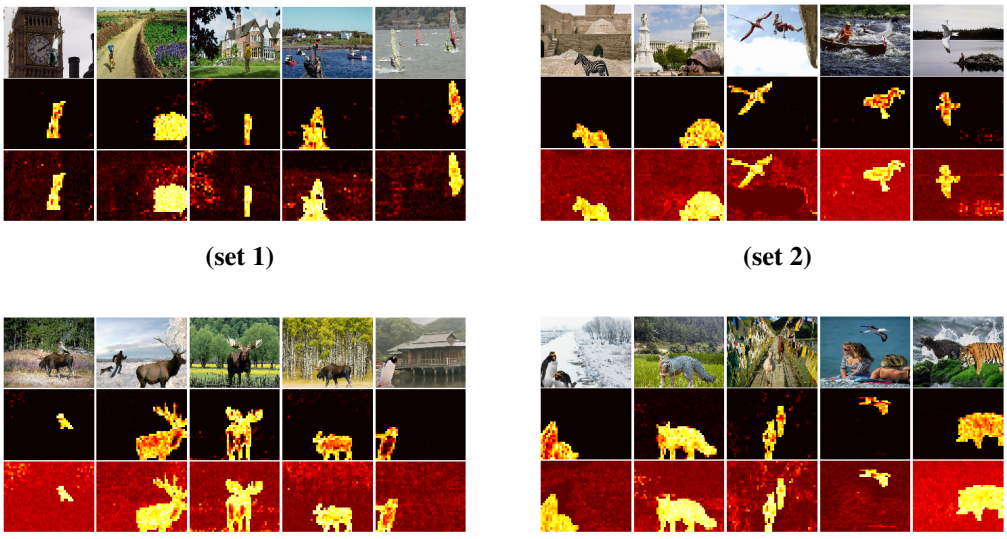


Figure 5.10: Example where forged region is detected partially.

Figure 5.11: Individual score maps (small), the global score map (big, top-left corner), and forged image and its ground truth (big, bottom-right corner).



(set 1) **(set 2)**

(set 3) **(set 4)**

Figure 5.12: Forged images from CASIA (first row), result maps of the proposed method (middle row), and results given by [113] (last row).

5.6 Conclusion

In this chapter, by analyzing the characteristics of JPEG compression in the DCT domain, we propose a new and efficient histogram-based solution for estimating the

quantization steps utilized in the first compression of a double-compressed JPEG image. The proposed solution performs precisely when the quantization step used in the first compression is greater than the double of the one used in the second compression. We also propose a simple and fast solution for exposing forgeries in compressed images. Given an image, assuming that it is double-compressed, we try to detect the quantization step used in the first compression for all the DCT coefficients. Having the first quantization step, we can create a model for the value of the doubly-quantized DCT coefficient. Blocks of which the DCT coefficients do not follow the given model are considered as forged. The proposed solution was tested on both simulated and real images. Forged images created from JPEG images and resaved under high quality JPEG can be detected correctly, and forged region can be localized precisely with the proposed method.

Chapter 6

Conclusion and Perspectives

Contents

6.1 Conclusion

By the advent of technologies, digital imaging has become a common and an important part of many information systems. It has been employed as the principal source of information in several applications, such as: industrial inspection, traffic monitoring, authentication, medical imaging, police investigation, and so on. However, along with these useful applications, there exist also different problems and challenges that we have to deal with to make these applications better. In this thesis, we have addressed to handle with some well-defined challenges related to digital imaging applications. In the chapter 3, we have handled with the presentation attack problem which is related to authentication systems using facial recognition. In fact, these systems are proven to be vulnerable to presentation attacks, where an attacker uses a presentation instrument such as photos, videos or masks of legitimate users' face to bypass and then gain access to the system. We have studied images' noise and found that, besides the acquisition noise, high-frequency information presenting on the surface of imaged objects also contributes to the noise that we can estimate from images. Acquisition noise depends mainly on the camera system. Meanwhile, the other one, referred as textural noise, behaves differently and dependently on the nature of imaged objects. By stabilizing the acquisition noise, we have proposed to study textural noise though out the estimated noise to detect face presentation attacks. A statistical model of noise local variance has been proposed to characterize noise's

behavior. Basing on the framework of hypothesis testing, we have exploited the proposed noise model to build a statistical test for face presentation attack detection. We have also created a new database for testing face PAD solutions. The proposed method has been tested on this database with a promising detection performance. QR codes have been popularly utilized in packaging for product identification and tracking via imaging devices such as smartphones. However, standard QR code is vulnerable to cloning attacks which prevent it from being served as a support for product authentication. In the chapter 4, we have introduced a watermarking solution to protect QR codes against cloning attacks. We have proposed to substitute white background of the standard QR codes by a particular random texture. The embedded texture, denoted as Clipping Gaussian Noise (CGN) texture, is proven to be sensitive with Print-and-Scan process. Any attempt for cloning the embedded QR codes would modify the statistical behavior of the texture and then can be detected. We have developed a statistical model for the variance of a sample of independent and identically distributed random variables. This model is then employed to model the DCT coefficients of the CGN texture. Different statistical detectors have been also developed basing on the proposed variance model and the framework of hypothesis testing. The proposed statistical detectors have been tested on both simulated and real data. The detection performance has been confirmed. The proposed watermarking solution makes QR code become a cheap and efficient support for product authentication.

In the chapter 5, we have concerned about the authenticity and the integrity of digital images. By analyzing the particular properties of DCT coefficients left by JPEG compression and double-compression, we have introduced a simple and efficient method for detecting image forgery and localizing forged region in images. For a given image, by assuming that it is recompensed after being forged and that the original image was also previously compressed, we supposed that the unforged region of the image is double-compressed. We have also supposed that the forged region's area is small enough and does not affect on the statistics of DCT coefficients extracted from all the 8×8-blocks of the image. Basing on the histogram of DCT coefficients and the quantization steps used in the second JPEG compression, we proposed a histogram-based solution to estimate the quantization steps which were used in the JPEG compression of the original image. Having the quantization steps of the first compression, we can then model precisely all the possible values of the DCT coefficients of blocks locating within the unforged region. Therefore, blocks of which DCT coefficient values do not follow the obtained model would be considered

as blocks within the forged region. We have studied the histogram-based algorithm for estimating quantization steps and the proposed image forgery detection solution with simulated data to understand the performance and also the limit of the proposed solutions. Our image forgery detection solution performs correctly when forged images are resaved in JPEG format with high Quality Factor (QF). The smaller the QF of the orginigal image is, the better the detection solution performs. We have also tested the proposed solution of images extracted from some public databases. Overall, in this thesis, we have addressed and proposed solutions for three different problems related to the drawbacks and limits of some precise digital imaging applications. The thesis contributes someway to the field of digital image forensics and statistical image modeling.

6.2 Perspectives

For the face PAD solution proposed in the chapter 3, we have not taken into account the aging and make-up problems, which can affect the detection performance strongly of the proposed solution. Further studies need to be considered to make the proposed solution more practical. For the watermarked QR codes proposed in the chapter 4, we have tested only codes printed on common A4 paper using a laser printer. In industrial packaging, other printing techniques and other types of paper have been employed. We have to adapt the proposed solution to fit these situations. We have to consider also further studies on the required resolution and size of valid watermarked QR codes. By the way, there may be other types of texture, not only the CGN one, which can be used to be embedded as a security level for QR codes. The proposed detection method is expected to be functional for some other types of texture. For both of the first two applications, studied images need to be taken under the same acquisition conditions. Therefore, we have to think about designing an acquisition system which is appropriated for each application. We may also think about other solutions which are less sensitive to acquisition noise. Working directly with the texture, not the image noise, can be an option. Studied images can be transformed into another domain, such as LBP, where texture information of the imaged object is amplified over the distortions caused by the acquisition process. The transformed image can be studied statistically in the spatial or frequential domain to be classified. In the chapter 5, we have proposed a efficient solution basing on the particular characteristics of DCT coefficients left by JPEG compression process. The solution permits us only detect forged images created and saved under certain

conditions, where JPEG compression left some signatures in the DCT coefficients. If forged images are created from uncrompress images or if they are resaved under JPEG format with a small quality factor, there will be no trace of JPEG compression or the difference between forged an unforged region in DCT domain would be vanished. The proposed solution would fail to detect forgery in these situations. We have to think about other solutions which can help us detect forgery from these images. Some directions can be considered:

- The CFA interpolation operations occurring during the demosaicing process would create spatial correlation between nearby pixels and possibly correlation between the three color channels. These correlations may differ between images coming from different sources. Forged images may be created from images of different sources. Inconsistency of the correlation characteristics between different regions of a same image may be considered as traces of forgery. There are many algorithms for CFA interpolation. We can try first to identify the interpolation algorithm employed for a given image. Studying image statistics in the frequency domain may be an option for that. Once we know the algorithm employed, we can model precisely the correlations between pixels and maybe between color channels. We can imagine proposing some test statistics basing on the obtained model to detect the inconsistent region of images.

- Image noise has been modeled and exploited for camera source identification problems in the works of Thai et al. [99, 100]. For forged images, forged an unforged regions may come from images captured by different cameras. Noise from these regions do not behave similarly. Therefore, we can also study image noise for image forgery detection. Thai et al. have proposed different statistical models for image noise. However, their models are only correct for homogenous regions in images. Therefore, we cannot use the same approach for the image forgery detection problem, where we have to care also the heterogenous regions of images. Thai et al. have demonstrated a relation between the expectation and the variance of image noise. There may also be certain relations between different other moments of the image noise, that we cannot model properly. These relations may be consistent over all an image, except for the forged region. Studying these relations, without modeling them, may permit us to have cues to detect forgery in high-quality images.

Appendix A

Appendix

Contents

A.1 Demonstration of theorem 1

We have that:

$$z_i - \overline{z} = \frac{63}{64} z_i - \frac{1}{64} \sum_{j \neq i} z_j. \tag{A.1}$$

We have that z_i is independent zero-mean and $Var(z_i) = f_\sigma(\mu_i)$. When μ_i is sufficient close to each other, $Var(z_i)$ is close enough to each other and Lindeberg's condition is satisfied. By invoking the Linderberg Central Limit Theorem, we have that:

$$z_i - \overline{z} \to \mathcal{N}(0, \sigma_i^2) \tag{A.2}$$

where

$$\sigma_i^2 = \frac{63^2}{64^2} f_\sigma(\mu_i) + \frac{1}{64^2} \sum_{j \neq i} f_\sigma(\mu_j) \tag{A.3}$$

So,

$$\tilde{z}_i = \frac{z_i - \overline{z}}{\sigma_i} \to \mathcal{N}(0, 1) \tag{A.4}$$

It follows that $\tilde{z}_i^2$ is a chi-square random variable with one d.f. Let denote, for all i, a random variable Y_i, defined as follows:

$$Y_i = \frac{1}{63}(z_i - \overline{z})^2 = \frac{\sigma_i^2}{63}\tilde{z}_i^2 \tag{A.5}$$

Denote $M_X(t)$ the Moment Generating Function of the random variable X. We have that:

$$M_{Y_i}(t) = \mathbb{E}(e^{Y_i t}) = \mathbb{E}(e^{\frac{\sigma_i^2}{63}\tilde{z}_i^2 t}) = M_{\tilde{z}_i^2}(\frac{\sigma_i^2}{63}t) = \left(1 - 2\frac{\sigma_i^2}{63}t\right)^{-\frac{1}{2}} \tag{A.6}$$

By comparing the MGF, it follows that:

$$Y_i \sim \mathcal{G}\left(\frac{1}{2}, \beta_i\right) \tag{A.7}$$

where

$$\beta_i = 2\frac{\sigma_i^2}{63} \tag{A.8}$$

We have that $S^2 = \sum_{i=0}^{63} Y_i$, is the sum of the correlated gamma random variable $Y_i, i \in \{0,,1,..,63\}$. It follows from [141] that the MGF of S^2 can be expressed as:

$$M_{S^2}(t) = [det(\mathbf{I}_{64} - t\mathbf{DC}]^{-\frac{1}{2}} \tag{A.9}$$

where $det(.)$ denotes the determinant operator, $\mathbf{I}_{64}$ is the 64x64 identity matrix, $\mathbf{D}$ is the 64x64 diagonal matrix with the entries β_i and $\mathbf{C}$ is the covariance matrix defined by:

$$\mathbf{C} = \begin{pmatrix} 1 & \sqrt{\rho_{1,2}} & ... & \sqrt{\rho_{1,64}} \\ \sqrt{\rho_{2,1}} & 1 & ... & \sqrt{\rho_{2,64}} \\ . & . & ... & . \\ \sqrt{\rho_{64,1}} & \sqrt{\rho_{64,2}} & ... & 1 \end{pmatrix} \tag{A.10}$$

with $\rho_{i,j}$ is the correlation coefficient between Y_i and Y_j within a block, i.e.,

$$\rho_{i,j} = \rho_{j,i} = \frac{Cov(Y_i, Y_j)}{\sqrt{Var(Y_i)Var(Y_j)}}. \tag{A.11}$$

Denoting $\{\lambda_i\}_{i=0}^{63}$ the eigenvalues of the matrix **DC**, we can rewrite the equation (A.9) as follow:

$$M_{S^2}(t) = \prod_{i=0}^{63}(1 - t\lambda_i)^{-\frac{1}{2}}. \tag{A.12}$$

The precise probability distribution of S^2 is given in , but it is so complicated mathematically. In this paper, we try to approximate it by a gamma distribution $\mathcal{G}(\alpha, \beta)$, which is much simpler to analyze. This approximation can be realized by using the moment matching method, which gives us that:

$$\alpha \approx \frac{(\sum_{i=0}^{63}\lambda_i)^2}{2\sum_{i=0}^{63}\lambda_i^2}, \quad \beta \approx \frac{\sum_{i=0}^{63}\lambda_i^2}{\sum_{i=0}^{63}\lambda_i}. \tag{A.13}$$

The λ_i values depend on the β_i and then depend on the block content. It follows that (α, β) is signal-dependent and:

$$\alpha\beta \approx \frac{1}{2}\sum_{i=0}^{63}\lambda_i \tag{A.14}$$

It's easy to show that the trace of a symmetric matrix is the sum of its eigenvalues. So that,

$$\sum_{i=0}^{63}\lambda_i = \sum_{i=0}^{63}\beta_i \tag{A.15}$$

Then, it follows from the equations (A.3) (A.8),(A.14) and (A.15) that:

$$\alpha\beta \approx \frac{1}{64}\sum_{i=1}^{64} f_\sigma(\mu_i) \tag{A.16}$$

A.2 Demonstration of the theorem 2

Note that:

$$z_i - \bar{z} = \frac{1}{n}\sum_{j\neq i}(z_i - z_j)$$

By invoking the Central Limit Theorem, the distribution of $z_i - \bar{z}$ can be approximated by a normal distribution. Indeed, we have that:

$$E(z_i - \bar{z}) = E(z_i) - E(\bar{z}) = 0$$

$$Var(z_i - \bar{z}) = Var(\frac{n-1}{n} z_i - \frac{1}{n}\sum_{j \neq i} z_j) = \frac{n-1}{n}\sigma_0^2$$

So that,

$$z_i - \bar{z} \to \mathcal{N}(0, \sqrt{\frac{n-1}{n}}\sigma_0)$$

Let $\sigma_1 = \sqrt{\frac{M-1}{M}}\sigma_0$, we have that:

$$\frac{z_i - \bar{z}}{\sigma_1} \to \mathcal{N}(0,1)$$

and so:

$$(\frac{z_i - \bar{z}}{\sigma_1})^2 \to \chi^2(1)$$

Denote $y = (\frac{z_i - \bar{z}}{\sigma_1})^2$, so we have:

$$\frac{1}{M-1}(z_i - \bar{z})^2 = \frac{\sigma_1^2}{M-1} y$$

The Moment Generating Function of $\frac{\sigma_1^2}{n-1} y$ is defined as follows:

$$M_{\frac{\sigma_1^2}{n-1} y}(t) = E(e^{t\frac{\sigma_1^2}{n-1} y}) = M_y(\frac{\sigma_1^2}{M-1} t) = (1 - \frac{2\sigma_1^2}{n-1} t)^{\frac{-1}{2}}$$

which is identical to the one of a random variable following a Gamma distribution whose the shape factor is $\frac{1}{2}$ and the scale factor is $\frac{2\sigma_1^2}{n-1} = \frac{2\sigma_0^2}{n}$. In recap, we have:

$$\frac{1}{n-1}(z_i - \bar{z})^2 \to \Gamma(\frac{1}{2}, \frac{2\sigma_0^2}{n})$$

Also by comparing the Moment Generating Function, it gives that the sum of Gamma random variables which have the same scale factor is also a Gamma random variable whose scale factor is identical to the latter and the shape factor is the sum of all previous shape factors. The theorem 2 is proven.

A.3 Wavelet-based denoising filter

This wavelet-based denoising technique was proposed by Lukas et al. in [97]. The denoising technique is proposed for grayscale 512×512 image. For larger images, it is proposed to process by blocks. For color images, the denoising algorithm is applied for each color channel separately. This denoising algorithm assumes that

the high-frequency wavelet coefficients of the noisy image are modeled as an additive mixture of a locally stationary i.i.d signal with zero mean and a stationary Gaussian noise $N(0, \sigma_0^2)$. There are many approaches to obtain an estimation of σ_0^2. In this work, we employed a fast noise variance estimation algorithm proposed by Immerkaer et al. in [142].

The wavelet-based denoising filter is built in two stages. The first stage consists to estimate the local image variances. In the second stage, we obtain an estimate of the denoised image in the wavelet domain by using the local Wiener filter. The whole

denoising algorithm is given below:

Data: 512×512 grayscale image

Result: Denoised image

Compute an estimation of σ_0, referred as $\hat{\sigma}_0$;

Compute the fourth-level wavelet decomposition of the noisy image with the 8-tap Daubechies Quadrature Mirror Filter (QMF);

for *each level* **do**

  Consider the three high-frequency (vertical, horizontal and diagonal) subbands;

  for *each subband,* denoted as s **do**

    Estimate the local variance of the original noise-free image for each wavelet coefficient $\hat{\sigma}^2(i,j)$, where $(i,j) \in J$ and J is an index set that depends on the decomposition level;

    for $W \in \{3,5,7,9\}$ **do**

      Using the Maximum à Posteriori (MAP) estimation with a square $W \times W$ neighborhood to obtain an estimation of $\hat{\sigma}^2(i,j)$;

$$\hat{\sigma}_W(i,j) = max\left(0, \frac{1}{W^2}\sum_{(i,j)\in J} s^2(i,j) - \sigma_0^2\right)$$

    end

    The value of $\hat{\sigma}(i,j)$ is obtained by:

$$\hat{\sigma}(i,j) = \min_{W\in\{3,5,7,9\}} \hat{\sigma}_W(i,j)$$

    Compute the denoised wavelet coefficients using the Wiener filter:

$$s_{denoised}(i,j) = s(i,j)\frac{\sigma^2(\hat{i},j)}{\sigma^2(\hat{i},j) + \hat{\sigma}_0^2}$$

  end

end

Final step: Apply the inverse wavelet transform to the denoised wavelet coefficients to obtain the denoised image;

Algorithm 1: Wavelet-based denoising algorithm.

A.4 Definition of Cumulant

In probability theory and statistics, the cumulants κ_n of a probability distribution are a set of quantities that provide an alternative to the moments of the distribution. The moments determine the cumulants in the sense that any two probability distributions whose moments are identical will have identical cumulants as well, and similarly the cumulants determine the moments.

The cumulant of a random variable X can be defined using the cumulant-generating function $K(t)$, which is the natural logarithm of the moment generating function $M(t)$ The two generating functions are defined as follows:

$$M(t) = \mathbf{E}\left[e^{tX}\right] \tag{A.17}$$

$$K(t) = logM(t) = log\mathbf{E}\left[e^{tX}\right] \tag{A.18}$$

The cumulant κ_n are obtained from the power series expansion of the cumulant generating function.

When X is a Gamma random variable, $X \sim \mathcal{G}(\alpha, \beta)$, its MGF is given by:

$$M(t) = (1 - \beta t)^{-\alpha} \tag{A.19}$$

Then its cululant generating function is derived as follows:

$$\begin{aligned} M(t) &= log\left((1-\beta t)^{-\alpha}\right) \\ &= -\alpha log(1-\beta t) \\ &= \alpha \sum_{n=1}^{\inf} \frac{(\beta t)^n}{n} \\ &= \sum_{n=1}^{\inf} \kappa_n \frac{t^n}{n!} \end{aligned} \tag{A.20}$$

So that, the n-th cumulant of a Gamma random variable is represented as follows:

$$\kappa_n = (n-1)!\alpha\beta^n \tag{A.21}$$

A.5 Demonstration of the Equation 3.31

We look to minimize Q, defined as below, with the linear constraint $B\theta = A$

$$Q = (h - w\theta)^T \hat{\Sigma}^{-1} (h - w\theta) \tag{A.22}$$

where $\hat{\theta}$ which minimize Q with no constraint is given as follows

$$\hat{\theta} = (w^T \hat{\Sigma}^{-1} w)^{-1} w^T \hat{\Sigma}^{-1} h \tag{A.23}$$

Let consider the Lagrangian function defined as follows

$$L(\theta, \lambda) = \frac{1}{2}(h - w\theta)^T \hat{\Sigma}^{-1} (h - w\theta) + \lambda^T (B\theta - A) \tag{A.24}$$

It is proposed actually to minimize this Lagrangian function. Denote $(\tilde{\theta}, \tilde{\lambda})$ the value which maximize $L(\theta, \lambda)$. The first order optimality conditions of the Lagrangian function give the following equations:

$$\frac{\partial L}{\partial \theta}(\tilde{\theta}, \tilde{\lambda}) = \quad w^T \hat{\Sigma}^{-1} w\tilde{\theta} - w^T \hat{\Sigma}^{-1} h + B\tilde{\lambda}^T \quad = 0 \tag{A.25}$$

$$\frac{\partial L}{\partial \lambda}(\tilde{\theta}, \tilde{\lambda}) = \quad B\tilde{\theta} - A \quad = 0 \tag{A.26}$$

From the first equation, we obtain:

$$\tilde{\theta} = \hat{\theta} - (w^T \hat{\Sigma}^{-1} w)^{-1} B^T \tilde{\lambda} \tag{A.27}$$

Replace in the second equation, we obtain:

$$B\hat{\theta} - B(w^T \hat{\Sigma}^{-1} w)^{-1} B^T \tilde{\lambda} = A \tag{A.28}$$

Then

$$\tilde{\lambda} = (B(w^T \hat{\Sigma}^{-1} w)^{-1} B^T)^{-1} (B\hat{\theta} - A) \tag{A.29}$$

And finally, we obtain the equation 3.31

$$\tilde{\theta} = \hat{\theta} - (w^T \hat{\Sigma}^{-1} w)^{-1} B^T (B(w^T \hat{\Sigma}^{-1} w)^{-1} B^T)^{-1} (B\hat{\theta} - A) \tag{A.30}$$

Appendix B

French Summary

Contents

B.1 Introduction

Depuis l'invention du tout premier appareil photo, on a commencé à capturer le monde réel en images. À l'époque des appareils photos argentiques, les images étaient stockées analogiquement sur des films argentiques. Les manipulations de ces images étaient très compliquées et demandaient énormément de savoir-faire professionnel. Les applications d'imagerie ne se sont limitées donc qu'au stockage et à des fins artistiques. Au cours des dernières décennies, l'avènement de la technologie d'imagerie numérique a radicalement changé notre façon de penser et de traiter des images. A la différence des images analogiques, les images numériques peuvent être facilement traitées et manipulées avec des logiciels informatiques. La puissance de calcul des systèmes informatiques nous permet aussi de traiter de manière automatique et rapide des données numériques. La transmission et la diffusion des images numériques sont aussi très faciles. Ce sont les raisons pour lesquelles de nombreuses applications d'imagerie numérique ont été développées, telles que: inspection industrielle, imagerie médicale, surveillance du trafic, authentification des personnes, identification des produits, etc. Dans cette thèse, on a abordé trois différents problèmes liés à trois applications précises d'imagerie numérique: authentification des personnes, authentification des produits et la détection des falsifications d'images numériques.

L'authentification par reconnaissance faciale devient actuellement une solution complémentaire permettant de renforcer la sécurité des systèmes d'information. Cependant, ces méthodes de reconnaissance faciale sont vulnérables aux attaques par usurpation d'identité. En effet, un attaquant peut contourner le processus d'authentification en présentant tout simplement, devant la caméra, une photo du visage d'un utilisateur légitime obtenue facilement sur les réseaux sociaux. Pour la sécurité de ces systèmes, il est donc vital de pouvoir identifier et éliminer toutes ces attaques. On propose une nouvelle méthode de détection de ces attaques basée sur l'étude du bruit des images. La section B.2 décrit en détail la méthode proposée.

Dans la section B.3, une méthode de tatouage numérique pour les QR codes est proposée. Les QR codes sont souvent imprimés sur les emballages des produits de consommation pour faciliter l'identification et l'acheminement des produits. On a proposé de tatouer numériquement des QR codes standards pour les rendre sensibles aux processus d'impression et de numérisation et pour que ces QR codes tatoués servent de moyen d'authentification pour les produits.

Dans la section B.4, on propose une solution simple mais efficace pour détecter

des falsifications dans des images compressées en format JPEG en étudiant des caractéristiques particulières des coefficients DCT . La section B.5 conclut ce chapitre.

B.2 Méthode de détection d'attaques par usurpation d'identité basée sur l'étude du bruit des images

Des bruits estimés à partir d'une image numérique se comportent en deux parties: les bruits d'acquisition et un autre, que l'on appelle le bruit des textures. Les bruits d'acquisition proviennent de différentes sources pendant le processus d'acquisition, telles que: le caractère non uniforme de la réponse du capteur, l'imperfection de la conversion analogique numérique, le caractère hétéroscédastique du bruit des photons, des pertes de compression, etc. Les bruits d'acquisition se lient fortement au système d'acquisition. Il y a déjà des travaux scientifiques [99, 115] qui étudiaient ces bruits pour identifier les sources des images. Le bruit des textures, quant à lui, est directement lié à la scène imagée. Ce sont des petites textures présentes à la surface de l'objet imagé qui peuvent être interprétées comme un bruit venant s'additionner au processus naturel de formation d'une image numérique. Pour les images venant de la reconnaissance faciale des systèmes d'authentification des personnes, ces textures peuvent être des mini-textures de peaux dans le cas non falsifié ou elles peuvent venir de l'imperfection du processus d'impression d'une photographie ou de l'effet moiré présent lors de l'utilisation d'un écran LCD au cours des attaques par usurpation.

Ainsi, le bruit des textures peut nous apporter des preuves des attaques par usurpation. Nous avons donc étudié ce bruit pour proposer une méthode de détection de ces attaques. Cependant, il est difficile d'étudier séparément le bruit des textures des bruits issus du processus d'acquisition. En maintenant statistiquement stable des bruits d'acquisition, on étudie le bruit des textures à travers les statistique du bruit estimées à partir de l'image.

B.2.1 Modélisation statistique

On a démontré que la variance locale du bruit, calculée à partir d'un bloc d'image de taille 8x8, suit approximativement une distribution Gamma si les espérances des pixels du bloc sont suffisamment proches entre elles, 3.2. Les paramètres de cette distribution Gamma sont dépendants du contenu du bloc. On note S^2, la variance

locale du bruit d'un bloc 8×8 donné, on a:

$$S^2 = \frac{1}{63}\sum_{i=1}^{64}(z_i - \bar{z})^2 \tag{B.1}$$

où i est l'indice des pixels du bloc, z_i la valeur du bruit estimé au pixel i, et $\bar{z}$ la moyenne du bruit du bloc concerné. On a démontré que

$$S^2 \sim \mathcal{G}(\alpha, \beta) \tag{B.2}$$

où α et β sont respectivement les paramètres de forme et d'échelle de la distribution Gamma. On a aussi obtenu que:

$$\alpha\beta \approx \frac{1}{64}\sum_{i=1}^{64} f_\sigma(\mu_i). \tag{B.3}$$

où μ_i est l'espérance du pixel i et $f_\sigma(.)$ est une fonction qui caractérise l'écart-type du bruit en fonction des espérances.

Le bruit des textures se comporte différemment entre différentes régions d'une image. En fait, la peau, les cheveux, les vêtements et surtout l'arrière-plan des images n'ont pas évidemment la même texture. Dans ce travail, on s'est donc intéressé seulement à la région de la peau, région présente dans toutes les images et qui couvre une partie suffisamment grande dans les images étudiées. On divise les images en blocs de taille 8×8 non-chevauchés. On s'intéresse seulement aux blocs situés dans la région de la peau. Après une opération de débruitage en utilisant des ondelettes, on peut décomposer chaque bloc en deux imagettes, qui sont aussi de taille 8×8 : le contenu et le bruit. Le contenu peut être associé aux espérances μ_i et le bruit est associé aux valeurs z_i dans les équations B.1 et B.3. Pour les blocs qui ont des contenus suffisamment proches entre eux, on peut supposer que les variances estimées S^2 suivent des distributions Gamma, $\mathcal{G}(\alpha, \beta)$, et qu'une relation linéaire entre α et β^{-1} existe.

En se basant sur le contenu des blocs, on divise les blocs de la région de la peau entre des groupes de niveau, que l'on appelle des *level-sets*, tels que les blocs qui appartiennent à un même groupe vont avoir des contenus suffisamment proches entre eux. En supposant que l'on divise les blocs en N groupes, on prend seulement les M ($M < N$) groupes dont la taille est la plus importante. Pour chaque bloc, on calcule la variance locale du bruit, donnée par l'équation B.1. En notant X_l, $l \in \{1, 2, ..M\}$ les M populations des variances locales du bruit issues des M level-sets sélectionnés,

on a

$$X_l \sim \mathcal{G}(\alpha_l, \beta_l) \tag{B.4}$$

où il existe des constantes (a_{l0}, b_{l0}) que $\alpha_l = a_{l0}/\beta_l + b_{l0}$ pour tout l. Sous forme matricielle, on a

$$\begin{bmatrix} 1 & -b_{10} & 0 & ... & ... \\ ... & ... & ... & ... & ... \\ ... & ... & 0 & 1 & -b_{M0} \end{bmatrix} \begin{bmatrix} \alpha_1\beta_1 \\ \beta_1 \\ ... \\ \alpha_M\beta_M \\ \beta_M \end{bmatrix} = \begin{bmatrix} a_{10} \\ ... \\ a_{M0} \end{bmatrix}. \tag{B.5}$$

En notant B, θ et A respectivement les trois termes de gauche à droite de l'équation B.5, cette équation devient tout simplement: $B\theta = A$.

Pour les images des visages réels, on suppose que l'équation B.5 est vérifiée. Pourtant, pour des images issues d'une attaque par usurpation d'identité, on n'a plus les mêmes textures sur la région de la peau, cette équation n'est donc plus vérifiée. Ainsi, le problème de la détection des attaques par usurpation d'identité revient à décider entre les deux hypothèses suivantes:

$$\begin{cases} \mathcal{H}_0 : B\theta = A \\ \mathcal{H}_1 : autrement. \end{cases} \tag{B.6}$$

B.2.2 Construction du test statistique

En notant le j-ième cumulant de la l-ième population par κ_{lj} (voir Appendix A.4 pour la définition des cumulants). On a

$$\kappa_{l1} = \alpha_l\beta_l, \tag{B.7}$$

$$\kappa_{lj} = (j-1)!\alpha_l\beta_l^j, \tag{B.8}$$

$$\kappa_{l,j+1}/\kappa_{lj} = j\beta_l. \tag{B.9}$$

On considère les mesures suivantes:

$$\eta_{l1} = \kappa_{l1}, \eta_{lj} = \kappa_{lj}/\kappa_{l,j-1}, \qquad j > 1 \tag{B.10}$$

$$\eta_l^T = (\eta_{l1}, \eta_{l2}, \eta_{l3}, \eta_{l4}) \quad l = 1, 2, ..M \tag{B.11}$$

$$\eta^T = [\eta_1^T, \eta_2^T, .., \eta_M^T]. \tag{B.12}$$

où X^T désigne la matrice transposée de la matrice X. On obtient aisément l'équation suivante:

$$\eta_l = \omega \begin{bmatrix} \alpha_l \beta_l \\ \beta_l \end{bmatrix}, \quad l = 1, 2.., M. \tag{B.13}$$

où

$$\omega^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix}. \tag{B.14}$$

Désignons $w = diag(\omega, \omega, .., \omega)$, obtenu en concaténant en diagonal M fois la matrice ω, on obtient la relation linéaire suivante:

$$\eta = w\theta. \tag{B.15}$$

Désignons Σ la matrice de covariance de η, et $\hat{\Sigma}$ une estimation de Σ et notons h l'échantillon de η. La meilleure estimation de θ doit minimiser la forme quadratique suivante:

$$Q = (h - w\theta)^T \hat{\Sigma}^{-1} (h - w\theta). \tag{B.16}$$

On note $\hat{\theta}$ et $\tilde{\theta}$ les valeurs de θ qui minimisent l'équation B.16 respectivement quand θ ne satisfait pas et satisfait l'équation B.5. En notant Q_0 et Q_1 respectivement les valeurs de Q quand θ est égal à $\hat{\theta}$ et $\tilde{\theta}$, et désignons $Q_{\mathcal{H}} = Q_0 - Q_1$, on a alors

$$Q_{\mathcal{H}} = (B\hat{\theta} - A)^T (B(w^T \hat{\Sigma}^{-1} w)^{-1} B^T)^{-1} (B\hat{\theta} - A). \tag{B.17}$$

La statistique $Q_{\mathcal{H}}$ est seuillée pour décider entre les hypothèses $\mathcal{H}_0$ et $\mathcal{H}_1$ [118]. On peut démontrer que sous l'hypothèse $\mathcal{H}_0$, $Q_{\mathcal{H}}$ suit une distribution χ^2 avec $r = rang(B)$ degrés de liberté. Quand la condition sous $\mathcal{H}_0$ n'est pas vérifiée, la distribution de $Q_{\mathcal{H}}$ est celle d'une variable aléatoire de la loi χ^2 non-centrée avec r degrés de liberté et le paramètre d'excentricité est définie par:

$$\Psi = (B\theta - A)^T (B(w^T \hat{\Sigma}^{-1} w)^{-1} B^T)^{-1} (B\theta - A). \tag{B.18}$$

Pour obtenir un test satisfaisant un taux de fausse alarme prescrit α_0, on décide de rejeter $\mathcal{H}_0$ quand $Q_{\mathcal{H}} \geq \chi^2_{r,\alpha_0}$ où $\chi^2_{r,\alpha_0} = \Phi^{-1}_{\chi^2_r}(1 - \alpha_0)$ et $\Phi^{-1}_{\chi^2_r}(.)$ est l'inverse de la distribution cumulée de la loi χ^2 avec r d.d.l.

B.2.3 Résultats

La méthode de détection basée sur la statistique $Q_{\mathcal{H}}$ a été testée sur des images simulées. On a obtenu de bonnes performances. Le figure B.1 donne la distribution du statistique $Q_{\mathcal{H}}$ pour différents ensembles des images simulées.

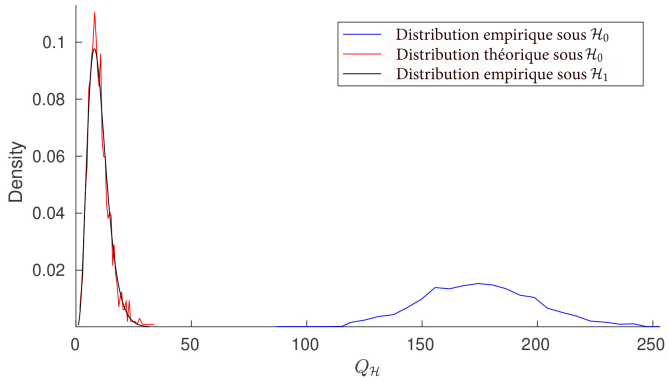
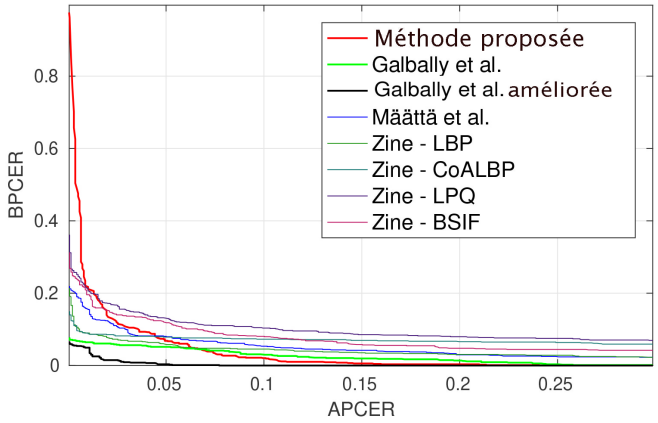
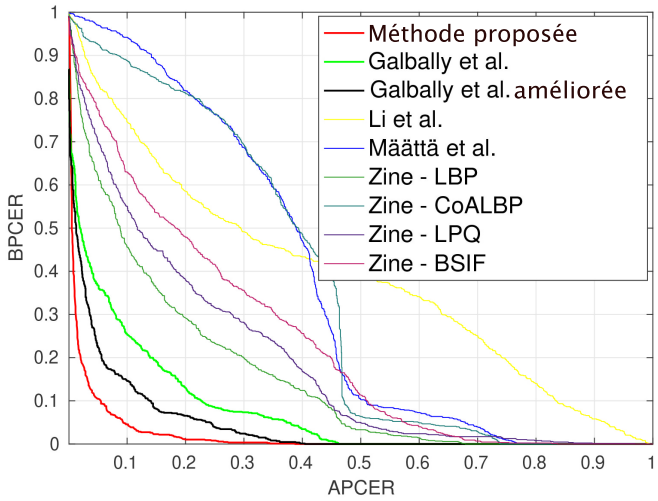


Figure B.1: Distribution du statistique $Q_{\mathcal{H}}$ pour différents ensembles des images simulées.

On a aussi testé la méthode proposée avec des images réelles. Vu que l'on a imposé des contraintes sur la stabilité des bruits d'acquisition pour modéliser le bruit des images dans notre méthode, on ne peut pas tester notre méthode sur les bases de données publiques existantes. On a donc créé une base de données, nommée DF-FPAD, contenant des images de vrais et faux visages capturées sous les mêmes conditions d'acquisition. On a testé la méthode proposée sur cette base de données. On a aussi comparé notre méthode avec certaines autres méthodes de la littérature: Galbally et al. [41], Li et al. [44], Määttä et al. [46], Boulkenafet et al. [8]. Un point fort de notre méthode est qu'elle n'a pas besoin à priori des images des attaques pour l'entrainement. Elle fonctionne correctement dans tout les cas. Les figures dans B.2 donnent les comparaisons des performances entre la méthode proposée et les autres méthodes citées précédemment. Quand un type d'attaque n'est pas présent dans la base d'entrainement, la performance des méthodes de la littérature est moins bonne que la nôtre. On a également ajouté la statistique $Q_{\mathcal{H}}$ comme un élément supplémentaire dans le vecteur de caractéristique d'une méthode utilisant des classificateurs SVM (Galbally et al. [41]). L'ajout de $Q_{\mathcal{H}}$ a permit d'améliorer nettement la performance de la méthode existante, figure B.2.



(a)



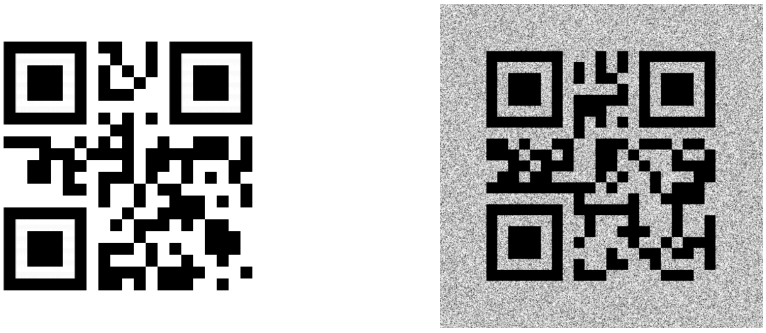
(b)

Figure B.2: La performance de la méthode proposée comparée avec celle des autres méthodes de littérature: Galbally et al. [41], Li et al. [44], Määttä et al. [46], Boulkenafet et al. [8]. Dans les figures, APCER (Attack Presentation Classification Error Rate) se signifie les Taux de Faux Positif, BPCER (Bona-fide Presentation Classification Error Rate) se signifie le Taux de Faux Négatif. Les courbes de performance de différentes approches de Boulkenafet et al. sont labellisées avec le préfixe *Zine-*. La courbe noire représente la performance de la méthode de Galbally et al. améliorée en ajoutant $Q_{\mathcal{H}}$ comme un élément supplémentaire dans le vecteur de caractéristique. (a) tous les types d'attaques sont présents dans la base d'entrainement; (b) certain types d'attaque ne sont pas présents dans la base d'entrainement.

B.3 Une solution de tatouage numérique pour sécuriser des QR codes

De nos jours, la contrefaçon de biens de consommation devient une préoccupation constante pour les consommateurs, les entreprises et la société. Des entreprises ont

développé de nombreuses technologies pour protéger leurs produits, telles que: des hologrammes, des dispositifs RFID ou NFC, des marqueurs biométriques, etc. Le coût de production de ces éléments de protection est relativement cher, surtout pour des produits à coût modique tels que des boîtes de médicament, des vêtements, etc. Une solution moins chère sera donc plus appréciée dans ce cas. Des QR codes ont déjà été imprimés sur l'emballage de produits de consommation pour servir à identifier le produit ou bien à tracer l'acheminement du produit. Pourtant, un QR code standard ne peut pas être utilisé pour authentifier des produits. En fait, les QR codes standards se composent tout simplement de modules blancs et noirs arrangés pour encoder un message. Ils sont faciles à copier et à reproduire. Dans l'objectif de transformer des QR codes en éléments pour authentifier des produits, on a proposé une solution de tatouage numérique pour protéger des QR codes contre des clonages reproductifs.



(a) **(b)**

Figure B.3: Exemple de QR code standard et QR code tatoué.

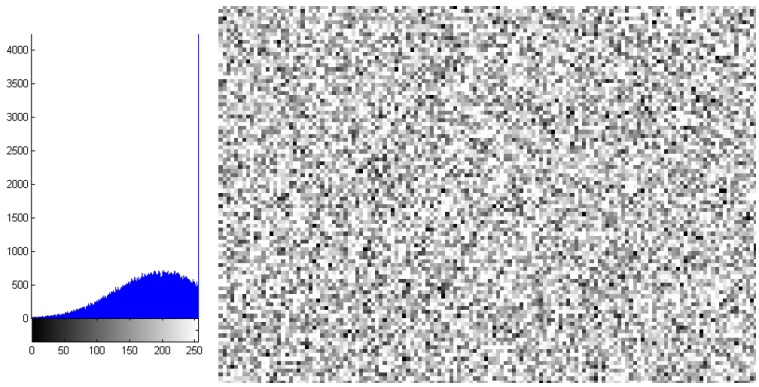


Figure B.4: CGN texture et son histogramme.

On propose de substituer l'arrière-plan blanc d'un QR code standard par un type de texture particulière pour créer des QR codes tatoués, Figure B.3. La texture que l'on a employée est nommée CGN, l'abréviation de Clipping Gaussian Noise

en anglais. C'est tout simplement du bruit gaussien pour lequel on remplace artificiellement des valeurs au dessous (resp. au dessus) d'un seuil par la valeur du seuil correspondant. La figure B.4 illustre la formation de la texture CGN. La texture CGN est sensible à des processus d'impression et de numérisation. Le caractère aléatoire de la texture CGN et des dégradations causées par le processus d'impression rendent des QR codes tatoués très difficile à reproduire par des contrefacteurs. La reproduction par numérisation et réimpression va dégrader la texture embarquée. En étudiant le comportement statistique de la texture embarquée, on peut vérifier l'authenticité des QR codes tatoués.

Les QR codes tatoués authentiques contiennent des textures CGN visuellement différentes mais statistiquement semblables. Dans le domaine fréquentiel, les textures CGN, pour une configuration fixe, se comportent statistiquement de manière identique. En fait, les coefficients DCT issus d'une fréquence donnée vont suivre exactement une même distribution statistique. Quand les QR codes tatoués sont imprimés, on suppose que des distorsions causées par le processus d'impression sont statistiquement identiques sur toute l'image, surtout sur des régions texturées. La version imprimée des textures CGN possède donc encore des caractéristiques statistiquement stables dans le domaine fréquentiel.

Pour authentifier un QR code tatoué à partir d'une image correctement cadrée, on extrait des textures CGN embarqués à partir de l'arrière-plan de ce dernier. Des zones extraites sont découpées en blocs de taille 8×8 non-chevauchés. On transforme ensuite chaque bloc en DCT. On obtient finalement 64 ensembles de coefficients correspondant aux 64 canaux DCT. Chaque ensemble contient des coefficients DCT issus d'un canal donné de tous les blocs étudiés. On peut considérer chaque ensemble comme un échantillon de variables aléatoires identiquement et indépendamment distribuées. La distribution des coefficients DCT est identique entre des images des QR codes authentiques. Par contre, des coefficient DCT des images des QR codes falsifiés ne se distribuent pas de la même façon que celles des images des codes authentiques. A partir de ces hypothèses et ces observations, on a proposé d'étudier des coefficients DCT des zones texturées pour authentifier des QR codes tatoués.

B.3.1 Modélisation statistique

Notant $\mathcal{I}$ l'ensemble des indices de tous les blocs de taille 8×8 situés dans la région texturée d'une image de QR code tatoué, N est la taille de $\mathcal{I}$, on note $\mathcal{Z}_j = \{z_{ji} | i \in$

$\mathcal{I}\}$ l'ensemble des coefficients DCT issues du canal $j, j \in \{1,2,..,64\}$. $\mathcal{Z}_j$ est un échantillon de variables aléatoires identiquement et indépendamment distribuées. La distribution statistique caractérisant $\mathcal{Z}_j$ est difficile à caractériser. Pour la simplicité de la conception des tests statistiques, on a décidé d'étudier seulement la variance de cette distribution. On découpe $\mathcal{Z}_j$ entre sous-ensembles de taille fixe. On estime la variance de la distribution à partir de chaque sous-ensemble. On a démontré que des variances estimées à partir de ces sous ensembles forme une population d'une variable aléatoire qui suit une loi Gamma dont le paramètre de forme est fixé à $1/2$ et le paramètre d'échelle est en fonction de la variance de la distribution caractérisant $\mathcal{Z}_j$. Supposons que l'on découpe $\mathcal{Z}_j$ en sous-ensembles de taille n, désignons $X_j = \{x_{ji} | i \in \mathcal{I}\}$, où les x_{ji} sont définies comme suit

$$x_{ji} = (z_{ji} - \bar{z}_{ji})^2, \forall i \in \mathcal{I} \tag{B.19}$$

où $\bar{z}_{ji}$ est la moyenne des valeurs du sous-ensemble qui contient z_{ji}. Dans la section 4.3.1, on a démontré que

$$x_i \sim \Gamma\left(\frac{1}{2}, (n-1)b_j\right) \tag{B.20}$$

où b_j est une constante caractérisant le canal j, elle peut être définie en fonction de la variance de la distribution qui caractérise l'ensemble $\mathcal{Z}_j$.

B.3.2 Tests basés sur un seul canal DCT

Si l'on prend seulement un canal DCT donné pour différencier les images des codes authentiques de celles des codes copiés, en ignorant l'indice du canal, et en notant $X = \{x_i | i \in \mathcal{I}\}$, le problème de détection peut être réécrit de la manière suivante:

$$\forall i \in \mathcal{I}, \begin{cases} \mathcal{H}_0 : \{x_i \sim \Gamma(a, b_0)\} \\ \mathcal{H}_1 : \{x_i \sim \Gamma(a, b_1)\}, b_1 \neq b_0 \end{cases} \tag{B.21}$$

où $a = \frac{1}{2}$, b_0 est connue, b_1 peut être connue ou inconnue.

Dans le contexte idéal où b_1 est connu, le rapport de vraisemblance (RV) des observations X_k est défini par:

$$\Lambda(X) = (\frac{1}{b_0} - \frac{1}{b_1}) \sum_{i \in \mathcal{I}} x_i. \tag{B.22}$$

Sous l'hypothèse $\mathcal{H}_h, h \in \{0,1\}$, on a

$$\Lambda(X) \sim \mathcal{N}(m_h, v_h) \tag{B.23}$$

où

$$m_h = Nab_h \frac{b_1 - b_0}{b_0 b_1}, \tag{B.24}$$

$$v_h = Nab_h^2 \frac{(b_1 - b_0)^2}{b_0^2 b_1^2}. \tag{B.25}$$

On défini le rapport de vraisemblance (RV) normalisé par:

$$\Lambda^\star(X) = \frac{\Lambda(X) - m_0}{\sqrt{v_0}}. \tag{B.26}$$

Par conséquence, le test RV $\delta^\star(X)$ basé sur le RV normalisé s'écrit comme suit:

$$\delta^\star(X) = \begin{cases} \mathcal{H}_0 \text{ si } \Lambda^\star(X) < \tau^\star \\ \mathcal{H}_1 \text{ si } \Lambda^\star(X) \geq \tau^\star \end{cases} \tag{B.27}$$

où le seuil de décision $\tau^\star$ est la solution de l'équation $\mathbb{P}_{\mathcal{H}_0}\left[\Lambda^\star(X) \geq \tau^\star\right] = \alpha_0$, et α_0 est le taux de fausse alarme prescrit. Le seuil de décision $\tau^\star$ et la puissance $\beta_{\delta^\star}$ sont donnés dans le théorème suivant:

Théorème 1 *Dans un contexte idéal où b_0 et b_1 sont tous connus, pour obtenir le taux de fausse alarme prescrit α_0, le seuil de décision et la puissance du test RV $\delta^\star(X)$ sont donnés par:*

$$\tau^\star = \Phi^{-1}(1 - \alpha_0) \tag{B.28}$$

$$\beta_{\delta^\star} = 1 - \Phi\left(\frac{m_0 - m_1 + \tau^\star\sqrt{v_0}}{\sqrt{v_1}}\right) \tag{B.29}$$

où $\Phi(.)$ et $\Phi^{-1}(.)$ désignent respectivement la distribution cumulée de la loi normale centrée réduite et sa fonction inverse.

Dans le contexte plus réaliste où b_1 est inconnu, afin de construire un test Rapport de Vraisemblance Généralisé (RVG), on estime d'abord b_1 par la méthode Maximum de Vraisemblance (MV) et puis on remplace b_1 par sa valeur estimée dans le RV. Le RVG des observations X_k est finalement défini par:

$$\hat{\Lambda}(X) = \frac{N}{b_0}\left[\bar{X}_k - ab_0 log(\bar{X}_k)\right] \tag{B.30}$$

où

$$\bar{X} = \frac{1}{N} \sum_{i \in \mathcal{I}} x_i$$

. Selon le théorème de la limite centrale, sous l'hypothèse $\mathcal{H}_h, h \in \{0,1\}$, on a

$$\bar{X} \to \mathcal{N}(\mu_h, \sigma_h^2) \tag{B.31}$$

où

$$\sigma_h = \sqrt{\frac{a}{N}} b_h \quad \text{and} \quad \mu_h = a b_h. \tag{B.32}$$

Notons $\bar{X}_h = \frac{\bar{X} - \mu_h}{\sigma_h}$, on obtient la version normalisée de $\bar{X}$

$$\bar{X}_h \to \mathcal{N}(0,1). \tag{B.33}$$

Ainsi, sous l'hypothèse $\mathcal{H}_h$, le RVG peut être récrit approximativement comme suit:

$$\hat{\Lambda}(X) \approx \frac{1}{2} \left(\bar{X}_h + c_h\right)^2 + d_h, \tag{B.34}$$

où

$$c_h = \sqrt{Na} \frac{b_h - b_0}{b_0}, \tag{B.35}$$

$$d_h = Na \frac{b_h - b_0 log(a b_h)}{b_0} - \frac{1}{2} c_h^2. \tag{B.36}$$

On a $c_0 = 0$, sous l'hypothèse $\mathcal{H}_0$, alors

$$\hat{\Lambda}^\star(X) = 2\left(\hat{\Lambda}(X) - d_0\right) \approx \bar{X}_0^2. \tag{B.37}$$

On trouve facilement que la distribution de $\hat{\Lambda}^\star(X)$ peut être approximée par celle d'une loi du χ^2 à un degré de liberté. Par conséquent, le test RVG $\hat{\delta}(X)$ basé sur le RVG transformé $\hat{\Lambda}^\star(X)$ s'écrit comme suit:

$$\hat{\delta}(X) = \begin{cases} \mathcal{H}_0 \text{ si } \hat{\Lambda}^\star(X) < \hat{\tau} \\ \mathcal{H}_1 \text{ si } \hat{\Lambda}^\star(X) \geq \hat{\tau} \end{cases} \tag{B.38}$$

où le seuil de décision $\hat{\tau}$ est la solution de l'équation $\mathbb{P}_{\mathcal{H}_0}\left[\hat{\Lambda}^\star(X)\right] = \alpha_0$, et α_0 est le taux de fausse alarme prescrit. Le seuil de décision $\hat{\tau}$ et la puissance $\beta_{\hat{\delta}}$ sont donnés dans le théorème suivant:

Théorème 2 *Dans le cas où b_1 est inconnu, pour obtenir le taux de fausse alarme prescrit α_0, le seuil de décision et la puissance du test RVG $\hat{\delta}(X)$ sont donnés par:*

$$\hat{\tau} = \Phi^{-1}_{\chi^2_1}(1-\alpha_0), \tag{B.39}$$

$$\beta_{\hat{\delta}} = 1 - \Phi_{\chi^2_{1,\lambda}}\left(\hat{\tau} + 2(d_0 - d_1)\right), \tag{B.40}$$

où $\Phi^{-1}_{\chi^2_1}(.)$ désigne l'inverse de la distribution cumulée de la loi du χ^2 d'un d.d.l, $\Phi_{\chi^2_{1,\lambda}}(.)$ désigne l'inverse de la distribution cumulée de la loi du χ^2 non-centré d'un d.d.l, dont le paramètre de décentralisation est $\lambda = c_1^2$.

B.3.3 Tests basés sur plusieurs canaux DCT

On peut aussi prendre en compte plusieurs canaux DCT pour construire des tests de détection plus puissants. Désignons $\mathcal{I}_{sub}$ l'ensemble des indices des canaux DCT employés pour la construction des tests. Le problème de détection peut être réécrit de la manière suivante:

$$\begin{cases} \mathcal{H}_0 : \left\{x_{ji} \sim \Gamma(a, b_{j0})\right\} \\ \mathcal{H}_1 : \left\{x_{ji} \sim \Gamma(a, b_{j1})\right\}, b_{j1} \neq b_{j0} \end{cases} \tag{B.41}$$

Dans le contexte idéal où les b_{j0}, b_{j1}, $\forall j \in \mathcal{I}_{sub}$, sont connus, le RV pour les observations X_j,$\forall j \in \mathcal{I}_{sub}$, est donné par:

$$\Lambda(X_{j,\forall j\in\mathcal{I}_{sub}}) = \sum_{j\in\mathcal{I}_{sub}} \left(\frac{1}{b_{j0}} - \frac{1}{b_{j1}}\right) \sum_{i\in\mathcal{I}} x_{ji} \tag{B.42}$$

Notons:

$$m_{jh} = Nab_{jh}\frac{b_{j1} - b_{j0}}{b_{j0}b_{j1}}, \tag{B.43}$$

$$v_{jh} = Nab_{jh}^2\frac{(b_{j1} - b_{j0})^2}{b_{j0}^2 b_{j1}^2}. \tag{B.44}$$

Sous l'hypothèse $\mathcal{H}_h$ on a:

$$\left(\frac{1}{b_{j0}} - \frac{1}{b_{j1}}\right) \sum_{i\in\mathcal{I}} x_{ji} \sim \mathcal{N}(m_{jh}, v_{jh}). \tag{B.45}$$

et puis:

$$\Lambda(X_{j,\forall j\in\mathcal{I}_{sub}}) \sim \mathcal{N}(m_h, v_h) \tag{B.46}$$

où $m_h = \sum_{j\in\mathcal{I}_{sub}} m_{jh}$, et $v_k = \sum_{j\in\mathcal{I}_{sub}} v_{jh}$. Définissons le RV normalisé comme suit

$$\Lambda^\star(X_{j,\forall j\in\mathcal{I}_{sub}}) = \frac{\Lambda(X_{j,\forall j\in\mathcal{I}_{sub}}) - m_0}{\sqrt{v_0}}. \tag{B.47}$$

Le test RV $\delta^\star(X)$ basé sur le RV normalisé s'écrit comme suit:

$$\delta^\star(X_{j,\forall j\in\mathcal{I}_{sub}}) = \begin{cases} \mathcal{H}_0 \text{ si } \Lambda^\star(X_{j,\forall j\in\mathcal{I}_{sub}}) < \tau^\star \\ \mathcal{H}_1 \text{ si } \Lambda^\star(X_{j,\forall j\in\mathcal{I}_{sub}}) \geq \tau^\star \end{cases} \tag{B.48}$$

où le seuil de décision $\tau^\star$ est la solution de l'équation

$$\mathbb{P}_{\mathcal{H}_0}\left[\Lambda^\star(X_{j,\forall j\in\mathcal{I}_{sub}}) \geq \tau^\star\right] = \alpha_0,$$

et α_0 est le taux de fausse alarme prescrit. Le seuil de décision $\tau^\star$ et la puissance $\beta_{\delta^\star}$ sont donnés dans le théorème suivant:

Théorème 3 *Dans un contexte idéal où les b_{j0} et b_{j1}, $j \in \mathcal{I}_{sub}$, sont tous connus, pour obtenir le taux de fausse alarme prescrit α_0, le seuil de décision et la puissance du test RV $\delta^\star(X_{j,\forall j\in\mathcal{I}_{sub}})$ sont donnés par:*

$$\tau^\star = \Phi^{-1}(1-\alpha_0) \tag{B.49}$$

$$\beta_{\delta^\star} = 1 - \Phi\left(\frac{m_0 - m_1 + \tau^\star\sqrt{v_0}}{\sqrt{v_1}}\right) \tag{B.50}$$

où $\Phi(.)$ et $\Phi^{-1}(.)$ désignent respectivement la distribution cumulée de la loi normale centrée réduite et sa fonction inverse.

Dans les contexte où les b_{j1} sont inconnues, le RVG des observation X_j, $j \in \mathcal{I}_{sub}$, est défini approximativement comme suit:

$$\hat{\Lambda}(X_{j,\forall j\in\mathcal{I}_{sub}}) \approx \sum_{j\in\mathcal{I}_{sub}} \frac{1}{2}\left(\bar{X}_{jh} + c_{jh}\right)^2 + \sum_{j\in\mathcal{I}_{sub}} d_{jh}, \tag{B.51}$$

où c_{jh} and d_{jh} sont des constantes définie par:

$$c_{jh} = \sqrt{Na}\frac{b_{jh} - b_{j0}}{b_{j0}}, \tag{B.52}$$

$$d_{jh} = Na\frac{b_{jh} - b_{j0}log(ab_{jh})}{b_{j0}} - \frac{1}{2}c_{jh}^2. \tag{B.53}$$

et $\bar{X}_{jh}$ sont des statistiques bien définies des observations X_j, elles suivent toutes la loi gaussienne centrée réduite. Notons que $c_{j0} = 0, \forall j \in \mathcal{I}_{sub}$, en posant

$$\hat{\Lambda}^\star(X_{j,\forall j\in\mathcal{I}_{sub}}) = 2\left(\hat{\Lambda}(X_{j,\forall j\in\mathcal{I}_{sub}}) - \sum_{j\in\mathcal{I}_{sub}} d_{j0}\right) \tag{B.54}$$

on a

$$\hat{\Lambda}^\star(X_{j,\forall j\in\mathcal{I}_{sub}}) \approx \sum_{j\in\mathcal{I}_{sub}} \bar{X}_{j0}^2. \tag{B.55}$$

Sous l'hypothèse $\mathcal{H}_h$, notons N_{sub} le nombre des canaux employés, on a:

$$\hat{\Lambda}^\star(X_{j,\forall j\in\mathcal{I}_{sub}}) \to \chi^2(N_{sub}) \tag{B.56}$$

On pourrait maintenant définir un test RVG comme suit:

$$\tilde{\delta}(X_{j,\forall j\in\mathcal{I}_{sub}}) = \begin{cases} \mathcal{H}_0 \text{ si } \hat{\Lambda}^\star(X_{j,\forall j\in\mathcal{I}_{sub}}) < \tilde{\tau} \\ \mathcal{H}_1 \text{ si } \hat{\Lambda}^\star(X_{j,\forall j\in\mathcal{I}_{sub}}) \geq \tilde{\tau} \end{cases} \tag{B.57}$$

où le seuil de décision $\tilde{\tau}$ est la solution de l'équation

$$\mathbb{P}_{\mathcal{H}_0}\left[\hat{\Lambda}^\star(X_{j,\forall j\in\mathcal{I}_{sub}}) \geq \tilde{\tau}\right] = \alpha_0,$$

où α_0 est le taux de fausse alarme prescrit. On obtient aussi le théorème suivant:

Théorème 4 *Dans un contexte idéal où les b_{j1}, $j \in \mathcal{I}_{sub}$, sont tous inconnus, pour obtenir le taux de fausse alarme prescrit α_0, le seuil de décision $\tilde{\tau}$ et la puissance du test RV $\tilde{\delta}(X_{j,\forall j\in\mathcal{I}_{sub}})$ sont donnés par:*

$$\tilde{\tau} = \Phi^{-1}_{\chi^2_{N_{sub}}}(1-\alpha_0), \tag{B.58}$$

$$\beta_{\hat{\delta}} = 1 - \Phi_{\chi^2_{N_{sub},\lambda}}\left(\tilde{\tau} + 2\sum_{j\in\mathcal{I}_{sub}}(d_{j0} - d_{j1})\right), \tag{B.59}$$

où $\Phi^{-1}_{\chi^2_{N_{sub}}}(.)$ désigne l'inverse de la distribution cumulée de la loi χ^2 de N_{sub} d.d.l, $\Phi_{\chi^2_{N_{sub},\lambda}}(.)$ désigne la distribution cumulée de la loi χ^2 non-centrée de N_{sub} d.d.l, dont le paramètre de décentralisation est donné par $\lambda = \sum_{j\in\mathcal{I}_{sub}} c_{j1}^2$.

B.3.4 Résultats

On a validé les tests proposés avec des données simulées. Les tests qui prennent en compte plusieurs canaux donnent des meilleurs résultats que ceux qui prennent un seul canal DCT. On a également testé la méthode de détection sur une petite base de données d'images réelles, que l'on a créée nous même. La détection est correcte, Figure B.5. Pourtant, il y a une petite perte de puissance. Cette perte peut être liée à la variation des conditions d'acquisition lors de la création de la base de données. On a aussi développé une application de démonstration pour la détection des codes tatoués non-authentiques. Dans des condition d'acquisition bien stables, l'application fonctionne correctement.

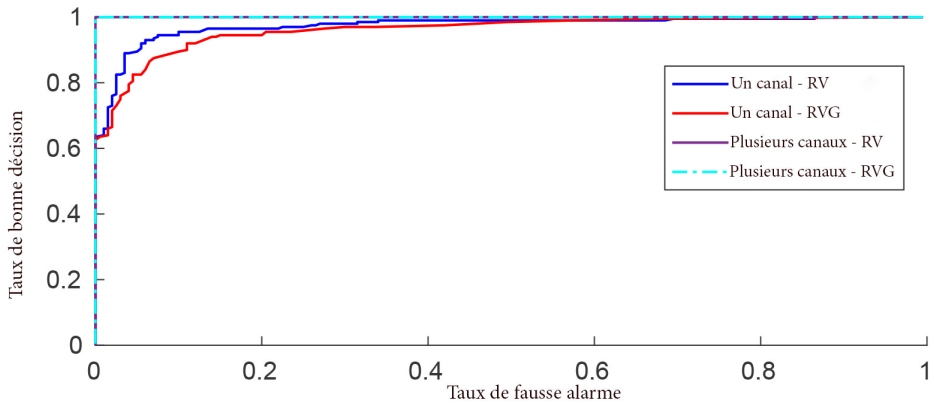


Figure B.5: Comparaison des performances entre différents tests sur une base de données réelle.

B.4 Méthode de détection des falsifications d'images numériques basée sur des signatures de la compression JPEG dans le domaine DCT

Le JPEG est un format d'image largement utilisé pour le stockage et la transmission des images numériques. Quand des images falsifiées sont créées, elles sont souvent sauvegardées sous le format JPEG sans ou avec l'intention des faussaires. En effet, des distorsions causées par la compression JPEG peuvent être utilisées efficacement pour masquer des traces de manipulation dans les images. Si l'image falsifiée est créée à partir d'images déjà compressées, la compression menée par le faussaire lors de la sauvegarde de leur produit final va provoquer une double-compression

JPEG dans l'image. La compression et la double-compression JPEG introduisent systématiquement dans les images certaines caractéristiques particulières dans le domaine DCT, ce que l'on peut exploiter pour proposer une méthode simple mais efficace pour détecter des falsifications d'images numériques. La figure B.6 illustre

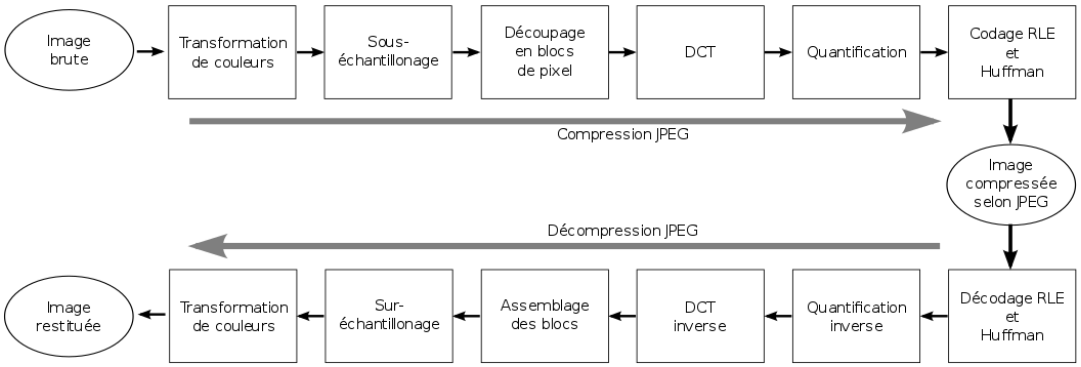


Figure B.6: Organigramme de la compression JPEG.

les différentes opérations de la compression et de la décompression JPEG. Après la transformation de couleurs et le sous-échantillonnage, l'image est découpée en blocs non-chevauchés de 8×8 pixels. Chaque bloc est transformé en DCT et puis quantifié en divisant chaque coefficient par des pas de quantification prédéfinis. Les valeurs des pas de quantification sont définies en fonction du facteur de qualité souhaitée de la compression JPEG. Les codages RLE et Huffman n'impactent pas la qualité de l'image. Les étapes de la décompression JPEG s'effectuent dans l'ordre inverse de la compression.

B.4.1 Caractéristiques des coefficients DCT quantifiés

Sans perte de généralité, pour un coefficient DCT donné, on note m, c, Q, d respectivement le coefficient DCT non-quantifié, sa valeur quantifiée, le pas de quantification associé et la valeur déquantifiée du coefficient DCT. On a:

$$c = \left[\frac{m}{Q}\right], \tag{B.60}$$

et

$$d = cQ = Q\left[\frac{m}{Q}\right]. \tag{B.61}$$

Lors d'une double compression JPEG, l'image est compressée deux fois. On note Q_i, c_i avec $i \in \{1,2\}$ respectivement le pas de quantification et la valeur quantifiée d'un coefficient DCT donné lors de la i-ème compression. On a la relation

suivante:

$$c_2 = \left[\frac{c_1 Q_1 + \mathfrak{E}}{Q_2}\right]. \tag{B.62}$$

où $\mathfrak{E}$ est une variable aléatoire qui représente des erreurs d'arrondis cumulées de différentes opérations survenues dans une double-compression JPEG. La valeur de $\mathfrak{E}$ est centrée autour de zéro. La probabilité d'être tombé dans l'intervalle $]-1,1[$ de $\mathfrak{E}$ est très élevée ($> 90\%$)[136]. On a

$$\left[\frac{c_1 Q_1}{Q_2}\right] - 1 \leq c_2 \leq \left[\frac{c_1 Q_1}{Q_2}\right] + 1. \tag{B.63}$$

et qu'il est fort probable que $c_2 = \left[\frac{c_1 Q_1}{Q_2}\right]$.

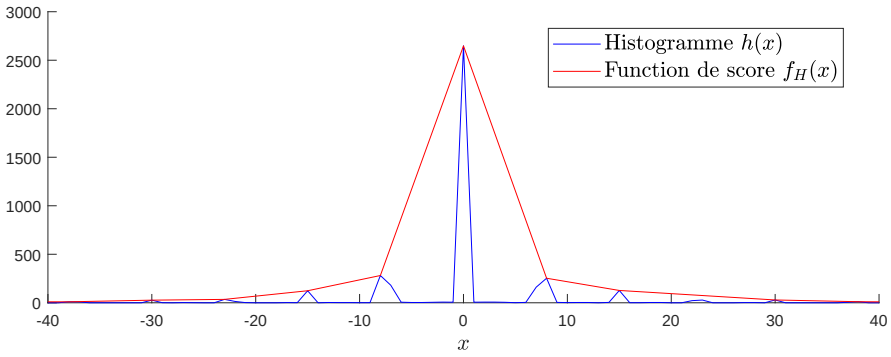
B.4.2 Méthode d'estimation du pas de quantification

Pour un couple de valeurs données de (Q_1, Q_2), à partir de l'équation B.62, on trouve que c_2 ne peut prendre que des valeur appartenant à un ensemble de valeurs prédéfinies. Cet ensemble peut être donc considéré comme la signature de la double quantification utilisant le couple (Q_1, Q_2). On est arrivé à proposer un algorithme pour estimer Q_1 sachant Q_2 en se basant sur l'histogramme des valeurs de c_2. On propose de faire varier Q_1 et de chercher exhaustivement la valeur de Q_1 qui minimise la mesure de différence donnée comme suit

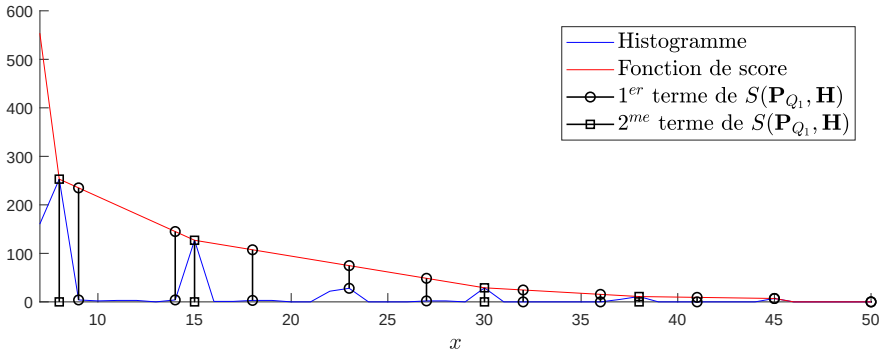
$$S(\mathbf{P}_{Q_1}, \mathbf{H}) = \sum_{p \in \mathbf{P}_{Q_1}} (f_H(p) - h(p)) + \sum_{p \in \mathbf{H}} (1 - e_{P_{Q_1}}(p)) f_H(p) \tag{B.64}$$

où les différents termes de cette équation sont définis comme suit:

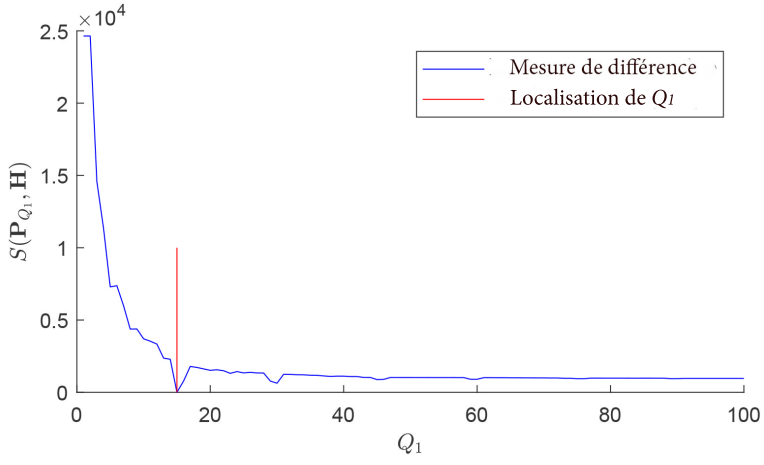
- $\mathbf{P}_{Q_1} = \{p_i, i \in \mathbf{I}_{P_{Q_1}}\}$, l'ensemble de toutes les valeurs possibles de c_2 sachant Q_1 et Q_2. $\mathbf{I}_{P_{Q_1}}$ est l'ensemble des indices de $\mathbf{P}_{Q_1}$.
- $\mathbf{H} = \{x_i, i \in \mathbf{I}_H\}$, l'ensemble qui désigne des localisations des pics de l'histogramme des valeurs de c_2. $\mathbf{I}_H$ est l'ensemble des indices de $\mathbf{H}$.
- $e_X(p) = 1$ if $p \in \mathbf{X}$ et $e_X(p) = 0$ if $p \notin \mathbf{X}$.
- $h(x)$, le nombre d'occurrences des valeurs de c_2 qui valent x.
- $f_H(x)$, une fonction de score qui est définie empiriquement à partir de l'histogramme des valeurs de c_2.



(a)



(b)



(b)

Figure B.7: Illustration de l'estimation de Q_1: (a) l'histogramme des coefficients DCT quantifiés et la fonction empirique de score; (b) l'illustration de différentes termes de la mesure de différence $S(\mathbf{P}_{Q_1}, \mathbf{H})$; (c) l'estimation de Q_1 en cherchant à minimiser $S(\mathbf{P}_{Q_1}, \mathbf{H})$.

Une illustration de l'estimation de Q_1 est donnée dans le Figure B.7.

L'algorithme proposée pour estimer Q_1 a été vérifié expérimentalement. On peut déterminer précisément Q_1 dans la plupart des cas quand $Q_1 > 2Q_2$, comme on peut voir dans la figure B.8, où on introduit des erreurs d'estimation de Q_1 pour

différentes valeurs de Q_1 et Q_2. Quand $Q_2 < 15$, on peut aussi déterminer correctement des valeurs Q_1 qui sont plus petites que $2Q_2$ et plus grand que Q_2. Quand $Q_1 = 2Q_2$ ou $Q_1 < Q_2$, l'algorithme ne fonctionne plus correctement. Plus la valeur de Q_2 est grande, plus la performance de l'algorithme baisse. Ainsi, si la deuxième compression est réalisée avec un facteur de qualité élevé (i.e Q_2 petit), il est fort probable de trouver des canaux DCT là où on peut estimer correctement les pas de quantification utilisés pendant la première compression.

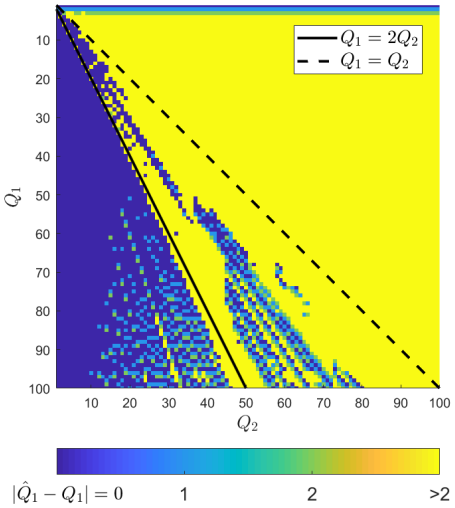


Figure B.8: La performance de la méthode d'estimation de Q_1 pour différentes valeurs de Q_2 visualisée à traverse de la valeur de $|\hat{Q}_1 - Q_1|$, où $\hat{Q}_1$ est la valeur estimée de Q_1.

B.4.3 Méthode de détection des falsifications

Pour un canal donné, une fois que l'on est arrivé à estimer correctement le pas de quantification utilisé pour la première compression, on peut en déduire l'ensemble de toutes les valeurs possibles des coefficients DCT, $\mathbf{P} = \{p_i, i \in \mathbf{I}_P\}$. S'il y a des blocs dont les coefficients DCT n'appartiennent pas à l'ensemble $\mathbf{P}$, ces blocs appartiennent à une région falsifiée. Dans cette étude, on a supposé que le nombre des blocs situés dans la région falsifiée de l'image est suffisamment petit pour que leurs coefficients DCT ne perturbent pas la forme globale de l'histogramme des coefficients DCT des blocs situés dans les régions non-falsifiées et que l'estimation du pas de quantification utilisé pendant la première compression est correcte. On note $\mathbf{Z} = \{z_i | i \in \mathbf{I}_B\}$ l'ensemble des coefficients DCT issus d'un canal donné, où I_B est l'ensemble qui désigne les indices de tous les blocs. On peut attribuer à chaque bloc un score qui représente un degré de falsification. Plus le score est élevé, plus le bloc

associé est issu de la région falsifiée. En notant $\mathbf{S} = \{s_i | i \in \mathbf{I}_B\}$ la matrice de score construite à partir d'un canal des coefficients DCT donné, on a

$$s_i = f_P(z_i). \tag{B.65}$$

où la fonction $f_P(z), z \in \mathrm{R}$, est définie comme suit:

$$f_P(z) = \begin{cases} d_P(z) & \text{if } d_P(z) > 1 \\ 0 & \text{if } d_P(z) \in \{0, 1\} \end{cases} \tag{B.66}$$

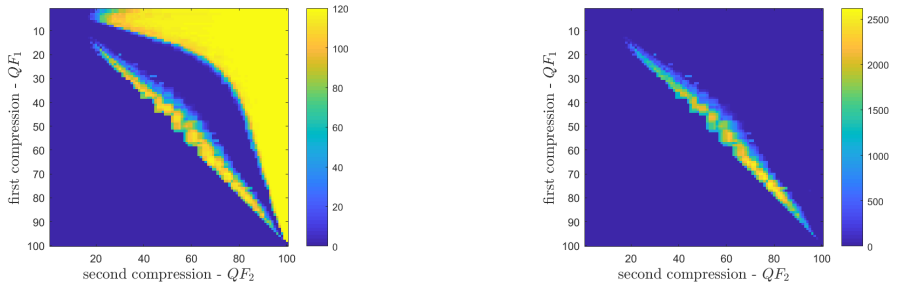
et $d_P(z)$ est la distance entre z et un élément de $\mathbf{P}$ qui est le plus proche de z. La fonction $f_P(z)$ prends des valeur de $d_P(z)$ quand $d_P(z) > 1$ parce que l'on a considéré tout simplement que plus $d_P(z)$ est grand, plus il est probable que le bloc associé soit dans la région falsifié. Quand $d_P(z) \leq 1$, il est fort probable que le bloc associé soit non-falsifié. On décide d'attribuer un score zéro dans ces cas pour minimiser le taux de fausse détection.

Ainsi, pour chaque canal DCT, là où on peut estimer Q_1, on peut construire une matrice de score pour servir à la détection des falsifications. On additionne toutes ces matrices pour obtenir finalement une matrice de score finale. La présence en groupement des scores non-nuls dans la matrice de score permet de mettre en relief les zones falsifiées dans l'image.

B.4.4 Résultats

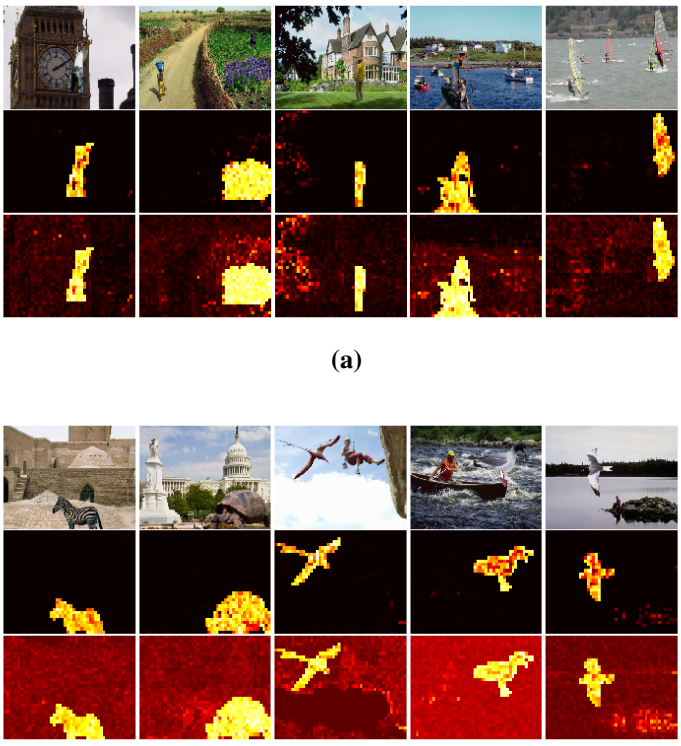
On a testé notre méthode de détection des falsifications avec des images simulées. Pour créer des images simulées, on commence par compresser deux images TIFF authentiques, I_1 et I_2, avec des facteur de qualité respectivement à QF_1 et 95. On copie une portion de taille fixe à partir d'I_2 et puis on la colle à un endroit précis défini au préalable dans I_1. Après cette manipulation, on sauvegarde I_1 sous format JPEG dont le facteur de qualité est QF_2 pour créer une image falsifiée simulée. On fait varier QF_1 et QF_2 indépendamment de 1 à 100 pour créer des images de tous les cas possibles de la double-compression JPEG. Notre méthode de détection fonctionne bien pour un grand nombre de configuration de (QF_1, QF_2), surtout quand QF_2 est grand (>95), Figure B.9. On a aussi testé notre méthode de détection avec des images issues de certaines bases de données publiques (CASIA 1.0 et CASIA 2.0). Parmi 177 images falsifiées étudiées, dont le facteur de qualité est supérieur à 95, notre

méthode a réussi de détecter et localiser correctement des zones falsifiées dans 169 images. Quelques exemples de résultat de détection sont donnés dans la Figure B.10.



(a) **(b)**

Figure B.9: La performance de la méthode de détection des falsifications évaluée avec des images simulées: (a) nombre de blocs falsifiés détectés et (b) nombre de blocs non-falsifiés mal classifiés pour différentes valeurs de facteurs de qualité employés pour la première (resp. second) compression JPEG QF_1 (resp. QF_2).



(a)

(b)

Figure B.10: Exemples de résultat de détection des falsifications d'images numériques (deuxième ligne) comparés avec ceux obtenus par la méthode [113] (troisième ligne).

B.5 Conclusion

Dans cette thèse, on avons traité différents problèmes concernant la vérification de l'authenticité et de l'intégrité des images numériques. Nous avons proposé des solutions pour détecter des falsifications des objets imagés pour des systèmes d'authentification des personnes par reconnaissance faciale et pour des systèmes d'authentification des produits via des QR code tatoués numériquement. Nous avons développé des approches de décision statistiques aussi fiables que possible permettant de garantir une probabilité de fausse alarme prescrite. A cet fin, les approches proposées consistent à concevoir des tests statistiques dans le cadre de la théorie des tests d'hypothèse basés sur des modèles paramétriques caractérisant des propriétés physiques ainsi que statistiques des objets imagés dans les images. On a aussi proposé une nouvelle solution pour la détection des falsifications d'images numériques en exploitant des caractéristiques particulières dans le domaine DCT de la compression JPEG.

Pour les applications d'authentification des personnes et des QR codes tatoués, on s'est basé sur des modèles statistiques des bruits et des textures. La variation des conditions d'acquisition peut affecter la qualité de ces modèles et conduire à des pertes de puissance des tests statistiques proposés. Nous devons concevoir des systèmes d'acquisition bien stabilisés appropriés pour chaque application afin d'atteindre la puissance théorique des tests. Sachant que l'étude des mini-textures à travers le bruit est trop sensible aux distorsions causées par l'acquisition, étudier directement des textures sans passer par le bruit est une piste intéressante à poursuivre. Il serait possible de transformer des images dans une autre espace où les distorsions causées par l'acquisition est minimisées (i.e. la transformation LBP). Ensuite, nous pourrons modéliser des images transformées dans le but de détecter la différence entre des images de différents objets. Une méthode par modèle serait envisageable.

La méthode proposée pour détecter des falsifications d'images numériques peut fonctionner correctement s'il y a des traces d'une double compression JPEG dans les images. Si les traces exploitées sont absentes ou trop faibles, la méthode proposée est inutile. Dans ce cas, il faudrait l'utiliser en complément d'autres méthodes de détection. Pour finir, l'exploitation des bruits de capteur ou les caractéristiques engendrées par le démosaiquage pour extraire des traces de falsification seraient des pistes très intéressantes à poursuivre.

Bibliography

[1] R. Ramanath, W. E. Snyder, Y. Yoo, and M. S. Drew, "Color image processing pipeline," *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 34–43, 2005-01. (cited on page 6, 10, 11)

[2] H. T. Sencar and N. Memon, *Digital Image Forensics: There is More to a Picture than Meets the Eye*. Springer-Verlag, 2013. (cited on page 11)

[3] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992-02. (cited on page 12, 12)

[4] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Comput. Surv.*, vol. 35, no. 4, pp. 399–458, 2003-12. (cited on page 15)

[5] M. Günther, L. El Shafey, and S. Marcel, "2d face recognition: An experimental and reproducible research survey," in *Idiap Publications*. (cited on page 15)

[6] M. M. Chakka, A. Anjos, S. Marcel, R. Tronci, D. Muntoni, G. Fadda, M. Pili, N. Sirena, G. Murgia, M. Ristori, F. Roli, J. Yan, D. Yi, Z. Lei, Z. Zhang, S. Z. Li, W. R. Schwartz, A. Rocha, H. Pedrini, J. Lorenzo-Navarro, M. Castrillón-Santana, J. Määttä, A. Hadid, and M. Pietikäinen, "Competition on counter measures to 2-d facial spoofing attacks," in *2011 International Joint Conference on Biometrics (IJCB)*, 2011-10, pp. 1–6. (cited on page 16, 16)

[7] S. Tirunagari, N. Poh, D. Windridge, A. Iorliam, N. Suki, and A. T. S. Ho, "Detection of face spoofing using visual dynamics," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 762–777, 2015-04. (cited on page 16, 16, 18)

[8] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face spoofing detection using colour texture analysis," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1818–1830, 2016-08. (cited on page 16, 16, 19, 61, 63, 64, 64, 64, 64, 127, 128)

[9] N. Erdogmus and S. Marcel, "Spoofing 2d face recognition systems with 3d masks," in *2013 International Conference of the BIOSIG Special Interest Group (BIOSIG)*, 2013-09, pp. 1–8. (cited on page 17)

[10] ——, "Spoofing face recognition with 3d masks," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 7, pp. 1084–1097, 2014-07. (cited on page 17)

[11] G. Tian, T. Mori, and Y. Okuda, "Spoofing detection for embedded face recognition system using a low cost stereo camera," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, 2016-12, pp. 1017–1022. (cited on page 17)

[12] X. Sun, L. Huang, and C. Liu, "Dual camera based feature for face spoofing detection," in *Pattern Recognition*, ser. Communications in Computer and Information Science. Springer, Singapore, 2016-11-05, pp. 332–344. (cited on page 17)

[13] I. Pavlidis and P. Symosek, "The imaging issue in an automatic face/disguise detection system," in *Proceedings IEEE Workshop on Computer Vision Beyond the Visible Spectrum: Methods and Applications*, 2000, pp. 15–24. (cited on page 17)

[14] Z. Zhang, D. Yi, Z. Lei, and S. Z. Li, "Face liveness detection by learning multispectral reflectance distributions," in *Face and Gesture 2011*, 2011-03, pp. 436–441. (cited on page 17)

[15] S. Bhattacharjee and S. Marcel, "What you can't see can help you - extended-range imaging for 3d-mask presentation attack detection," in *2017 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–7. (cited on page 17, 17)

[16] L. Sun, W. Huang, and M. Wu, "TIR/VIS correlation for liveness detection in face recognition," in *Computer Analysis of Images and Patterns*, ser. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2011-08-29, pp. 114–121. (cited on page 17)

[17] R. Raghavendra, K. B. Raja, and C. Busch, "Presentation attack detection for face recognition using light field camera," *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 1060–1075, 2015-03. (cited on page 17)

[18] A. Sepas-Moghaddam, L. Malhadas, P. L. Correia, and F. Pereira, "Face spoofing detection using a light field imaging framework," *IET Biometrics*, vol. 7, no. 1, pp. 39–48, 2018. (cited on page 17)

[19] R. Raghavendra, K. B. Raja, S. Venkatesh, F. A. Cheikh, and C. Busch, "On the vulnerability of extended multispectral face recognition systems towards presentation attacks," in *2017 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*, 02 2017, pp. 1–8. (cited on page 17)

[20] K. T. Nguyen, F. Retraint, C. Zitzmann, F. Morain-Nicolier, A. Delahaies, and H. P. Nguyen, "Face spoofing detection for smartphones using a 3d reconstruction and the motion sensors," in *2018 4th International Conference on Information Systems Security and Privacy (ICISSP18)*, 2018. (cited on page 17)

[21] K. Kollreider, H. Fronthaler, M. I. Faraj, and J. Bigun, "Real-time face detection and motion analysis with application in "liveness" assessment," *Trans. Info. For. Sec.*, vol. 2, no. 3, pp. 548–558, 2007-09. (cited on page 18)

[22] E. S. Ng and A. Y. S. Chia, "Face verification using temporal affective cues," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, 2012-11, pp. 1249–1252. (cited on page 18)

[23] R. W. Frischholz and A. Werner, "Avoiding replay-attacks in a face recognition systenm using head-pose estimation," in *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, ser. AMFG '03. IEEE Computer Society, 2003, pp. 234–235. (cited on page 18)

[24] M. D. Marsico, M. Nappi, D. Riccio, and J. L. Dugelay, "Moving face spoofing detection via 3d projective invariants," in *2012 5th IAPR International Conference on Biometrics (ICB)*, 2012-03, pp. 73–78. (cited on page 18)

[25] T. Wang, J. Yang, Z. Lei, S. Liao, and S. Z. Li, "Face liveness detection using 3d structure recovered from a single camera," in *2013 International Conference on Biometrics (ICB)*, 2013-06, pp. 1–6. (cited on page 18)

[26] K. Kollreider, H. Fronthaler, and J. Bigun, "Verifying liveness by multiple experts in face biometrics," in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2008-06, pp. 1–6. (cited on page 18)

[27] M. Killioğlu, M. Taşkiran, and N. Kahraman, "Anti-spoofing in face recognition with liveness detection using pupil tracking," in *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 2017-01, pp. 000 087–000 092. (cited on page 18)

[28] A. Ali, F. Deravi, and S. Hoque, "Directional sensitivity of gaze-collinearity features in liveness detection," in *2013 Fourth International Conference on Emerging Security Technologies*, 2013-09, pp. 8–11. (cited on page 18)

[29] G. Chetty, "Liveness verification in audio-video speaker authentication," in *Proceedings of the 10th Australian International Conference on Speech Science and Technology (SST'04)*, 2004, pp. 358–363. (cited on page 18)

[30] S. Bharadwaj, T. I. Dhamecha, M. Vatsa, and R. Singh, "Computationally efficient face spoofing detection with motion magnification," in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013-06, pp. 105–110. (cited on page 18)

[31] G. Pan, Z. Wu, and L. Sun, "Liveness detection for face recognition," *Recent Advances in Face Recognition*, pp. 109–124, 2008. (cited on page 18)

[32] W. Bao, H. Li, N. Li, and W. Jiang, "A liveness detection method for face recognition based on optical flow field," in *International Conference on Image Analysis and Signal Processing, 2009. IASP 2009*, 2009-04, pp. 233–236. (cited on page 18)

[33] K. Kollreider, H. Fronthaler, and J. Bigun, "Non-intrusive liveness detection by face images," *Image and Vision Computing*, vol. 27, no. 3, pp. 233–244, 2009-02-02. (cited on page 18)

[34] S. Kim, S. Yu, K. Kim, Y. Ban, and S. Lee, "Face liveness detection using variable focusing," in *2013 International Conference on Biometrics (ICB)*, jun. 2013, pp. 1–6. (cited on page 18)

[35] S. Kim, Y. Ban, and S. Lee, "Face liveness detection using defocus," *Sensors (Basel, Switzerland)*, vol. 15, no. 1, pp. 1537–1563, 2015-01-14. (cited on page 18)

[36] Q. T. Phan, D. T. Dang-Nguyen, G. Boato, and F. G. B. D. Natale, "FACE spoofing detection using LDP-TOP," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016-09, pp. 404–408. (cited on page 18)

[37] S. R. Arashloo, J. Kittler, and W. Christmas, "An anomaly detection approach to face spoofing detection: A new formulation and evaluation protocol," *IEEE Access*, vol. 5, pp. 13 868–13 882. (cited on page 18)

[38] Y. Tian and S. Xiang, "Detection of video-based face spoofing using LBP and multiscale DCT," in *Digital Forensics and Watermarking*, ser. Lecture Notes in Computer Science. Springer, Cham, pp. 16–28. (cited on page 18)

[39] A. Anjos, M. M. Chakka, and S. Marcel, "Motion-based counter-measures to photo attacks in face recognition," *IET Biometrics*, vol. 3, no. 3, pp. 147–158, 2014-09. (cited on page 18)

[40] J. Yan, Z. Zhang, Z. Lei, D. Yi, and S. Z. Li, "Face liveness detection by exploring multiple scenic clues," in *2012 12th International Conference on Control Automation Robotics Vision (ICARCV)*, 2012-12, pp. 188–193. (cited on page 18)

[41] J. Galbally and S. Marcel, "Face anti-spoofing based on general image quality assessment," in *2014 22nd International Conference on Pattern Recognition*, 2014-08, pp. 1173–1178. (cited on page 19, 62, 62, 63, 64, 127, 127, 128)

[42] A. P. S. Bhogal, D. Söllinger, P. Trung, and A. Uhl, "Non-reference image quality assessment for biometric presentation attack detection," in *2017 5th International Workshop on Biometrics and Forensics (IWBF)*, 2017-04, pp. 1–6. (cited on page 19)

[43] O. Nikisins, A. Mohammadi, A. Anjos, and S. Marcel, "On effectiveness of anomaly detection approaches against unseen presentation attacks in face anti-spoofing." (cited on page 19)

[44] J. Li, Y. Wang, T. Tan, and A. K. Jain, "Live face detection based on the analysis of Fourier spectra," in *Biometric Technology for Human Identification*, vol. 5404, 2004, pp. 296–303. (cited on page 19, 63, 64, 127, 128)

[45] X. Tan, Y. Li, J. Liu, and L. Jiang, "Face liveness detection from a single image with sparse low rank bilinear discriminative model," in *Computer Vision*

– *ECCV 2010*, ser. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2010-09-05, pp. 504–517. (cited on page 19, 55)

[46] J. Määttä, A. Hadid, and M. Pietikäinen, "Face spoofing detection from single images using micro-texture analysis," in *2011 International Joint Conference on Biometrics (IJCB)*, 2011-10, pp. 1–7. (cited on page 19, 61, 63, 64, 87, 127, 128)

[47] J. Maatta, A. Hadid, and M. Pietikainen, "Face spoofing detection from single images using texture and local shape analysis," *IET Biometrics*, vol. 1, no. 1, pp. 3–10, 2012-03. (cited on page 19)

[48] M. Farmanbar and n. Toygar, "Spoof detection on face and palmprint biometrics," *Signal, Image and Video Processing*, vol. 11, no. 7, pp. 1253–1260. (cited on page 19)

[49] H. K. Bashier, S. H. Lau, P. Y. Han, L. Y. Ping, and C. M. Li, "Face spoofing detection using local graph structure," in *Proceedings of the 2014 International Conference on Computer, Communications and Information Technology*. Atlantis Press, 2014. (cited on page 19)

[50] M. Abdullah, M. Sayeed, K. Sonai Muthu, H. Bashier, A. Azman, and S. Ibrahim, "Face recognition with symmetric local graph structure (SLGS)," *Expert Systems with Applications*, vol. 41, no. 14, pp. 6131–6137, oct. 2014. (cited on page 19)

[51] P. Chiang, N. Khanna, A. K. Mikkilineni, M. V. O. Segovia, S. Suh, J. P. Allebach, G. T. Chiu, and E. J. Delp, "Printer and scanner forensics," vol. 26, no. 2, pp. 72–83. (cited on page 23, 23, 23, 24)

[52] C.-Y. Lin and S.-F. Chang, *Distortion Modeling and Invariant Extraction for Digital Image Print-and-Scan Process*. (cited on page 26)

[53] K. Solanki, U. Madhow, B. S. Manjunath, S. Chandrasekaran, and I. El-Khalil, "'print and scan' resilient data hiding in images," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 4, pp. 464–478. (cited on page 26, 84)

[54] G. N. Ali, A. K. Mikkilineni, P.-J. Chiang, J. P. Allebach, G. T.-C. Chiu, and E. J. Delp, "Intrinsic and extrinsic signatures for information hiding and secure printing with electrophotographic devices." (cited on page 27)

[55] Y. Wu, X. Kong, X. You, and Y. Guo, "Printer forensics based on page document's geometric distortion," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, pp. 2909–2912. (cited on page 27)

[56] O. Bulan, J. Mao, and G. Sharma, "Geometric distortion signatures for printer identification," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1401–1404. (cited on page 28)

[57] S. Shang and X. Kong, "Printer and scanner forensics," in *Handbook of Digital Forensics of Multimedia Data and Devices*. Wiley-Blackwell, pp. 375–410. (cited on page 28)

[58] C. H. Lampert, L. Mei, and T. M. Breuel, "Printing technique classification for document counterfeit detection," in *2006 International Conference on Computational Intelligence and Security*, vol. 1, pp. 639–644. (cited on page 28)

[59] M. Umadevi, A. Agarwal, and C. Raghavendra Rao, "Printed text characterization for identifying print technology using expectation maximization algorithm," in *Multi-disciplinary Trends in Artificial Intelligence*, ser. Lecture Notes in Computer Science, C. Sombattheera, A. Agarwal, S. K. Udgata, and K. Lavangnananda, Eds. Springer Berlin Heidelberg, pp. 201–212. (cited on page 28)

[60] C. Schulze, M. Schreyer, A. Stahl, and T. Breuel, "Using DCT features for printing technique and copy detection," in *Advances in Digital Forensics V*, ser. IFIP Advances in Information and Communication Technology, G. Peterson and S. Shenoi, Eds. Springer Berlin Heidelberg, pp. 95–106. (cited on page 28)

[61] E. Kee and H. Farid, "Printer profiling for forensics and ballistics," in *Proceedings of the 10th ACM Workshop on Multimedia and Security*, ser. MM&Sec '08. ACM, pp. 3–10. (cited on page 28)

[62] J. Gebhardt, M. Goldstein, F. Shafait, and A. Dengel, "Document authentication using printing technique features and unsupervised anomaly detection," in *2013 12th International Conference on Document Analysis and Recognition*, pp. 479–483. (cited on page 28)

[63] M. Embar, L. F. McHugh, IV, and W. R. Wesselman, "Printer watermark obfuscation," in *Proceedings of the 3rd Annual Conference on Research in Information Technology*, ser. RIIT '14. ACM, pp. 15–20. (cited on page 28)

[64] S. Ibrahim, M. Afrakhteh, and M. Salleh, "Adaptive watermarking for printed document authentication," in *5th International Conference on Computer Sciences and Convergence Information Technology*, pp. 611–614. (cited on page 28)

[65] A. L. Varna, S. Rane, and A. Vetro, "Data hiding in hard-copy text documents robust to print, scan and photocopy operations," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1397–1400. (cited on page 29)

[66] J. Picard, "Digital authentication with copy-detection patterns," in *Proc. SPIE*, vol. 5310, 2004, pp. 176–183. (cited on page 29)

[67] A. T. P. Ho, B. A. M. Hoang, W. Sawaya, and P. Bas, "Document authentication using graphical codes: reliable performance analysis and channel optimization," *EURASIP Journal on Information Security*, vol. 2014, no. 1, p. 9, 2014-12-01. (cited on page 29, 69)

[68] C. Baras and F. Cayre, "2d bar-codes for authentication: A security approach," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, 2012-08, pp. 1760–1766. (cited on page 29)

[69] I. Tkachenko, W. Puech, C. Destruel, O. Strauss, J. M. Gaudin, and C. Guichard, "Two-level QR code for private message sharing and document authentication," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 3, pp. 571–583, Mars 2016. (cited on page 29)

[70] Y. M. Wang, C. T. Sun, P. C. Kuan, C. S. Lu, and H. C. Wang, "Secured graphic QR code with infrared watermark," in *2018 IEEE International Conference on Applied System Invention (ICASI)*, 2018-04, pp. 690–693. (cited on page 30)

[71] N. Teraura, "Counterfeit detection by smartphone using double-encoded two-dimensional code," in *Innovative Mobile and Internet Services in Ubiquitous Computing*, ser. Advances in Intelligent Systems and Computing. Springer, Cham, 2017-07-10, pp. 455–466. (cited on page 30)

[72] S. Vongpradhip and S. Rungraungsilp, "QR code using invisible watermarking in frequency domain," in *2011 Ninth International Conference on ICT and Knowledge Engineering*, pp. 47–52. (cited on page 30)

[73] T. V. Bui, N. K. Vu, T. T. P. Nguyen, I. Echizen, and T. D. Nguyen, "Robust message hiding for QR code," in *2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 520–523. (cited on page 30)

[74] P. Lin, Y. Chen, E. J. Lu, and P. Chen, "Secret hiding mechanism using QR barcode," in *2013 International Conference on Signal-Image Technology Internet-Based Systems*, pp. 22–25. (cited on page 30)

[75] A. J. Fridrich, B. D. Soukal, and A. J. Lukáš, "Detection of copy-move forgery in digital images," in *in Proceedings of Digital Forensic Research Workshop*, 2003. (cited on page 35, 35)

[76] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," 2004. (cited on page 35)

[77] X. Kang and S. Wei, "Identifying tampered regions using singular value decomposition in digital image forensics," in *2008 International Conference on Computer Science and Software Engineering*, vol. 3, 2008-12, pp. 926–930. (cited on page 35)

[78] L. Kang and X. p. Cheng, "Copy-move forgery detection in digital image," in *2010 3rd International Congress on Image and Signal Processing*, vol. 5, 2010-10, pp. 2419–2421. (cited on page 35)

[79] H.-J. Lin, C.-W. Wang, and Y.-T. Kao, "Fast copy-move forgery detection," *WSEAS Trans. Sig. Proc.*, vol. 5, no. 5, pp. 188–197, 2009-05. (cited on page 35)

[80] J. Zhang, Z. Feng, and Y. Su, "A new approach for detecting copy-move forgery in digital images," in *2008 11th IEEE Singapore International Conference on Communication Systems*, 2008-11, pp. 362–366. (cited on page 36)

[81] W. Luo, J. Huang, and G. Qiu, "Robust detection of region-duplication forgery in digital image," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 4, 2006-08, pp. 746–749. (cited on page 36)

[82] S. D. Lin and T. Wu, "An integrated technique for splicing and copy-move forgery image detection," in *2011 4th International Congress on Image and Signal Processing*, vol. 2, 2011-10, pp. 1086–1090. (cited on page 36)

[83] M. Ghorbani, M. Firouzmand, and A. Faraahi, "DWT-DCT (QCD) based copy-move image forgery detection," in *2011 18th International Conference on Systems, Signals and Image Processing*, 2011-06, pp. 1–4. (cited on page 36)

[84] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 758–767, 2005-02. (cited on page 36)

[85] B. Mahdian and S. Saic, "Blind authentication using periodic properties of interpolation," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 529–538, 2008-09. (cited on page 36)

[86] M. Kirchner, "Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue," in *Proceedings of the 10th ACM Workshop on Multimedia and Security*, ser. MM&Sec '08. ACM, 2008, pp. 11–20. (cited on page 36)

[87] Z. Zhang, Y. Zhang, Z. Zhou, and J. Luo, "Boundary-based image forgery detection by fast shallow CNN," *arXiv:1801.06732 [cs]*, 2018-01-20. (cited on page 36)

[88] V. Conotter, G. Boato, and H. Farid, "Detecting photo manipulation on signs and billboards," in *2010 IEEE International Conference on Image Processing*, 2010-09, pp. 1741–1744. (cited on page 37)

[89] M. K. Johnson and H. Farid, "Exposing digital forgeries by detecting inconsistencies in lighting," in *Proceedings of the 7th Workshop on Multimedia and Security*, ser. MM&Sec '05. ACM, 2005, pp. 1–10. (cited on page 37)

[90] ——, "Exposing digital forgeries in complex lighting environments," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 450–461, 2007-09. (cited on page 37)

[91] ——, "Exposing digital forgeries through specular highlights on the eye," in *Proceedings of the 9th International Conference on Information Hiding*, ser. IH'07. Springer-Verlag, 2007, pp. 311–325. (cited on page 37)

[92] W. Zhang, X. Cao, J. Zhang, J. Zhu, and P. Wang, "Detecting photographic composites using shadows," in *2009 IEEE International Conference on Multimedia and Expo*, 2009-06, pp. 1042–1045. (cited on page 37)

[93] M. K. Johnson and H. Farid, "Exposing digital forgeries through chromatic aberration," in *Proceedings of the 8th Workshop on Multimedia and Security*, ser. MM&Sec '06. ACM, 2006, pp. 48–55. (cited on page 38)

[94] O. Mayer and M. C. Stamm, "Accurate and efficient image forgery detection using lateral chromatic aberration," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1762–1777, 2018-07. (cited on page 38)

[95] I. Yerushalmy and H. Hel-Or, "Digital image forgery detection based on lens and sensor aberration," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 71–91, 2011-03-01. (cited on page 39)

[96] S. Lyu, "Estimating vignetting function from a single image for image authentication," in *Proceedings of the 12th ACM Workshop on Multimedia and Security*, ser. MM&Sec '10. ACM, 2010, pp. 3–12. (cited on page 39)

[97] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, 2006-06. (cited on page 39, 116)

[98] 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, 2008-03. (cited on page 39)

[99] T. H. Thai, R. Cogranne, and F. Retraint, "Camera model identification based on the heteroscedastic noise model," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 250–263, 2014-01. (cited on page 39, 44, 56, 112, 123)

[100] T. H. Thai, F. Retraint, and R. Cogranne, "Camera model identification based on the generalized noise model in natural images," *Digital Signal Processing*, vol. 48, pp. 285–297. (cited on page 39, 85, 112)

[101] P. Kakar, N. Sudha, and W. Ser, "Exposing digital image forgeries by detecting discrepancies in motion blur," *IEEE Transactions on Multimedia*, vol. 13, no. 3, pp. 443–452, 2011-06. (cited on page 39)

[102] A. C. Popescu and H. Farid, "Exposing digital forgeries in color filter array interpolated images," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 3948–3959, 2005-10. (cited on page 39)

[103] A. C. Gallagher and T. Chen, "Image authentication by detecting traces of demosaicing," in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2008-06, pp. 1–8. (cited on page 39)

[104] A. E. Dirik and N. Memon, "Image tamper detection based on demosaicing artifacts," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, 2009-11, pp. 1497–1500. (cited on page 39)

[105] Z. Lin, R. Wang, X. Tang, and H.-Y. Shum, "Detecting doctored images using camera response normality and consistency," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005-06, pp. 1087–1092 vol. 1. (cited on page 40)

[106] Y. Hsu and S. Chang, "Camera response functions for image forensics: An automatic algorithm for splicing detection," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 816–825, 2010-12. (cited on page 40)

[107] S. Agarwal and H. Farid, "Photo forensics from JPEG dimples," in *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, 2017-12, pp. 1–6. (cited on page 40)

[108] S. Ye, Q. Sun, and E. Chang, "Detecting digital image forgeries by measuring inconsistencies of blocking artifact," in *2007 IEEE International Conference on Multimedia and Expo*, 2007-07, pp. 12–15. (cited on page 40)

[109] W. Li, Y. Yuan, and N. Yu, "Passive detection of doctored JPEG image via block artifact grid extraction," *Signal Processing*, vol. 89, no. 9, pp. 1821–1829, 2009-09-01. (cited on page 40)

[110] B. Liu, C.-M. Pun, and X.-C. Yuan, "Digital image forgery detection using JPEG features and local noise discrepancies," *The Scientific World Journal*, vol. 2014, 2014. (cited on page 40)

[111] Z. Ting and W. Rangding, "Doctored JPEG image detection based on double compression features analysis," in *2009 ISECS International Colloquium on Computing, Communication, Control, and Management*, vol. 2, 2009-08, pp. 76–80. (cited on page 40)

[112] W. Junwen, L. Guangjie, D. Yuewe, and W. Zhiquan, "Detecting JPEG image forgery based on double compression," *Journal of Systems Engineering and Electronics*, vol. 20, no. 5, pp. 1096–1103, 2009-10. (cited on page 40)

[113] Z. Lin, J. He, X. Tang, and C.-K. Tang, "Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis," *Pattern Recognition*, vol. 42, no. 11, pp. 2492–2501, 2009-11-01. (cited on page 40, 106, 106, 107, 143)

[114] T. Yamasaki, T. Matsunami, and K. Aizawa, "Detecting resized JPEG images by analyzing high frequency elements in DCT coefficients," in *2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2010-10, pp. 567–570. (cited on page 40)

[115] T. Hai Thai, F. Retraint, and R. Cogranne, "Generalized signal-dependent noise model and parameter estimation for natural images," *Signal Processing*, vol. 114, pp. 164–170, 2015-09. (cited on page 44, 53, 85, 123)

[116] Y. Ban, S. Kim, K. Toh, and S. Lee, "Face detection based on skin color likelihood," *Pattern Recognition*, vol. 47, no. 4, pp. 1573–1585, 2014. (cited on page 46)

[117] P. Billingsley, *Probability and measure*. Wiley, google-Books-ID: Q2IPAQAAMAAJ. (cited on page 47)

[118] J. Gurland and R. C. Tripathi, "Test of hypotheses in some families of discrete distributions," *Bulletin of the Greek Mathematical Society*, vol. 19, no. 19, pp. 217–239, 1978. (cited on page 53, 53, 126)

[119] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical poissonian-gaussian noise modeling and fitting for single-image raw-data," *IEEE Transactions on Image Processing*, vol. 17, no. 10, pp. 1737–1754, 2008-10. (cited on page 53)

[120] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *2012 BIOSIG Proceedings of the International Conference of Biometrics Special Interest Group (BIOSIG)*, 2012, pp. 1–7. (cited on page 55)

[121] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li, "A face antispoofing database with diverse attacks," in *2012 5th IAPR International Conference on Biometrics (ICB)*, 2012-03, pp. 26–31. (cited on page 55)

[122] D. Wen, H. Han, and A. Jain, " Face Spoof Detection with Image Distortion Analysis," *IEEE Trans. Information Forensic and Security*, vol. 10, no. 4, pp. 746–761, April 2015. (cited on page 55)

[123] B. Gianmarco, S. Riccardo, N. F. Igor, T. Aris, and C. Enrico, "Survey of techniques for fight against counterfeit goods and intellectual property rights (ipr) infringing," EU Science Hub - The European Commission's science and knowledge service, Tech. Rep., 2015. (cited on page 68)

[124] J. Dittmann, L. C. Ferri, and C. Vielhauer, "Hologram watermarks for document authentications," in *Proceedings International Conference on Information Technology: Coding and Computing*, 2001-04, pp. 60–64. (cited on page 68)

[125] M. O. Lehtonen, F. Michahelles, and E. Fleisch, "Trust and security in RFID-based product authentication systems," *IEEE Systems Journal*, vol. 1, no. 2, pp. 129–144, 2007-12. (cited on page 68)

[126] B. Song and C. J. Mitchell, "RFID authentication protocol for low-cost tags," in *Proceedings of the First ACM Conference on Wireless Network Security*, ser. WiSec '08. ACM, pp. 140–147. (cited on page 68)

[127] C. E. Turcu, C. O. Turcu, M. Cerlinca, T. Cerlinca, R. Prodan, and V. Popa, "An RFID-based system for product authentication," in *Eurocon 2013*, 2013-07, pp. 32–39. (cited on page 68)

[128] N. Alzahrani and N. Bulusu, "Securing pharmaceutical and high-value products against tag reapplication attacks using NFC tags," in *2016 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2016-05, pp. 1–6. (cited on page 68)

[129] T. Hongthai and D. Thanapatay, "The development of encrypted near field communication data exchange format transmission in an NFC passive tag for checking the genuine product," in *2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2017-06, pp. 889–894. (cited on page 68)

[130] H. Nam, K. Song, D. Ha, and T. Kim, "Inkjet-printing-based structural coloring for anti-counterfeit applications," in *2015 Transducers - 2015 18th International Conference on Solid-State Sensors, Actuators and Microsystems (TRANSDUCERS)*, 2015-06, pp. 1417–1420. (cited on page 68)

[131] ISO/IEC, "Information Technology—Automatic Identification and Data Capture Techniques—Bar Code Symbology—QR Code," International Organization for Standardization, ISO/IEC 18004:2000, Mars 2000. (cited on page 69)

[132] A. E. Dirik and B. Haas, "Copy detection pattern-based document protection for variable media," *IET Image Processing*, vol. 6, no. 8, pp. 1102–1113, November 2012. (cited on page 71)

[133] T. Thai, R. Cogranne, and F. Retraint, "Statistical model of quantized DCT coefficients: Application in the steganalysis of jsteg algorithm," *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 1980–1993, 2014. (cited on page 73)

[134] T. Qiao, F. Retraint, R. Cogranne, and C. Zitzmann, "Steganalysis of JSteg algorithm using hypothesis testing theory," *EURASIP Journal on Information Security*, vol. 2015, no. 1, p. 2, 2015-03-13. (cited on page 73)

[135] J. Neyman and E. S. Pearson, "IX. on the problem of the most efficient tests of statistical hypotheses," *Phil. Trans. R. Soc. Lond. A*, vol. 231, no. 694, pp. 289–337. (cited on page 75)

[136] T. H. Thai, R. Cogranne, F. Retraint, and T. N. C. Doan, "JPEG quantization step estimation and its applications to digital image forensics," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 123–133, 2017-01. (cited on page 96, 96, 139)

[137] W. Luo, J. Huang, and G. Qiu, "JPEG error analysis and its applications to digital image forensics," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 480–491, 2010-09. (cited on page 96)

[138] T. Pevny and J. Fridrich, "Detection of double-compression in JPEG images for applications in steganography," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 2, pp. 247–258, 2008-06. (cited on page 96)

[139] T. H. Thai, R. Cogranne, and F. Retraint, "Steganalysis of jsteg algorithm based on a novel statistical model of quantized DCT coefficients," in *2013 IEEE International Conference on Image Processing*, 2013-09, pp. 4427–4431. (cited on page 97)

[140] J. Dong, W. Wang, and T. Tan, "CASIA image tampering detection evaluation database," in *2013 IEEE China Summit and International Conference on Signal and Information Processing*, 2013-07, pp. 422–426. (cited on page 106)

[141] M. Alouini, A. Abdi, and M. Kaveh, "Sum of gamma variates and performance of wireless communication systems over nakagami-fading channels,"

IEEE Transactions on Vehicular Technology, vol. 50, no. 6, pp. 1471–1480, nov. 2001. (cited on page 114)

[142] J. Immerkær, "Fast noise variance estimation," in *Computer Vision and Image Understanding*, vol. 64, 09 1996, pp. 300–302. (cited on page 117)