



HAL
open science

Statistical Modeling for Detection of Digital Image Forgery

Thi Ai Nhàn Lê

► **To cite this version:**

Thi Ai Nhàn Lê. Statistical Modeling for Detection of Digital Image Forgery. Image Processing [eess.IV]. Université de Technologie de Troyes, 2021. English. NNT : 2021TROY0046 . tel-03810789

HAL Id: tel-03810789

<https://theses.hal.science/tel-03810789v1>

Submitted on 11 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Thèse
de doctorat
de l'UTT

Thi Ai Nhàn LÊ

Statistical Modeling for Detection of Digital Image Forgery

Champ disciplinaire :
Sciences pour l'Ingénieur

2021TROY0046

Année 2021

THESE
pour l'obtention du grade de
DOCTEUR
de l'UNIVERSITE DE TECHNOLOGIE DE TROYES
en SCIENCES POUR L'INGENIEUR

Spécialité : OPTIMISATION ET SURETE DES SYSTEMES

présentée et soutenue par

Thi Ai Nhàn LÊ

le 16 décembre 2021

Statistical Modeling for Detection of Digital Image Forgery

JURY

M. Frédéric MORAIN-NICOLIER	PROFESSEUR DES UNIVERSITES	Président
M. Jean-Luc DUGELAY	PROFESSEUR MINES-TELECOM - HDR	Rapporteur
M. William PUECH	PROFESSEUR DES UNIVERSITES	Rapporteur
M. Tong QIAO	ASSOCIATE PROFESSOR	Examineur
Mme Cathel ZITZMANN	ENSEIGNANTE CHERCHEURE EPF	Examinatrice
M. Florent RETRAINT	PROFESSEUR DES UNIVERSITES	Directeur de thèse

I would like to dedicate this thesis to my loving family.

Acknowledgements

This work has been carried out within the Computer Science and Digital Society laboratory at the Troyes University of Technology (UTT). It was funded by the REGION GRAND EST/CD10 (Conseil Départemental) under the Projet IDEO, and was partially supported by the National Research Agency (ANR) Project DEFACTO under Project ANR-16-DEFA-0002.

First and foremost, I would like to express my deep gratitude and my greatest respect to my supervisor, Professor Florent RETRAINT, for his continuous support and guidance throughout my four years in UTT. He has accompanied with me from my Master's internship and encouraged me to continue discovering this attractive research field through this doctoral project. His professional expertise and understanding have considerably added to my graduate experience. It was a great pleasure for me to work with him and an honor to be his student.

I would like to express my special thanks to Professor William PUECH and Professor Jean-Luc DUGELAY for accepting to review my PhD thesis. I address my gratitude to Professor Frédéric MORAIN-NICOLIER, Doctor Cathel ZITZMANN and Doctor Tong QIAO for agreeing to examine this thesis.

Thanks to all my colleagues who gave me company during my graduate school, especially Hoai Phuong NGUYEN, Yanli CHEN and Gaël MAHFOUDI.

Finally, I am grateful to my husband, parents, sisters and brothers for their support, love and patience that provided me with the strength to overcome all the difficult moments.

LÊ Thị Ái Nhàn

Abstract

In today's digital age, the trustworthiness of image content is of great concern due to the dissemination of easy-to-use and low-cost image editing tools. Forged images can be used to manipulate public opinion during elections, commit fraud, discredit or blackmail people. Faced with such a serious situation, we develop in this doctoral project three versatile techniques based on (i) *demosaiicing traces* (ii) *JPEG-compression traces*, and (iii) *resampling traces* for detecting forged digital images and localizing various types of tampering therein. Although these techniques are different, they work under the common assumption that manipulations may alter some underlying statistical properties of natural images. A two-steps detection process has been adopted for every detection technique: (i) analyze and model statistical features of both the authentic and forged images associated with specific in-camera and/or post-camera traces, then (ii) design a statistical detector to differentiate between the authentic and forged images by estimating statistical changes in their models. Various numerical experiments on several well-known benchmark datasets highlight the performances and robustness of the proposed detection techniques.

Résumé

À l'ère du numérique, la crédibilité du contenu des images est une préoccupation majeure en raison de la popularité des outils d'édition faciles à utiliser et peu coûteux. Les images falsifiées peuvent être utilisées pour manipuler l'opinion publique lors des élections, commettre des fraudes et discréditer ou faire chanter des personnes. Face à cette situation préoccupante, nous développons dans cette thèse trois techniques efficaces basées sur (i) *les traces de dématricage* (ii) *les traces de compression JPEG*, et (iii) *les traces de rééchantillonnage* pour détecter les images falsifiées et localiser les différents types de falsification. Bien que ces techniques soient différentes, elles fonctionnent sous l'hypothèse commune que les manipulations peuvent altérer certaines propriétés statistiques sous-jacentes des images naturelles. Un processus de détection en deux étapes a été adopté pour chaque technique de détection : (i) analyser et modéliser les caractéristiques statistiques des images authentiques et falsifiées, puis (ii) concevoir un détecteur statistique pour différencier les images falsifiées des images authentiques en estimant les changements dans leurs modèles. Diverses expérimentations numériques sur plusieurs ensembles de données de référence bien connus mettent en évidence la qualité des performances et la robustesse des techniques de détection proposées.

Contents

Contents	v
1 General Introduction	1
1.1 General Context	1
1.2 Research Framework	3
1.3 Thesis Outline	4
1.4 Publications	5
2 Overview on Digital Image Processing Pipeline and Passive Forgery Detection	7
2.1 Introduction	7
2.2 Image Processing Pipeline in Digital Still Cameras	7
2.2.1 From Image Scene to RAW Image	8
2.2.2 From RAW Image to TIFF Image	10
2.2.3 From TIFF Image to JPEG Image	13
2.3 Techniques of Digital Image Forgery	19
2.3.1 Cloning	19
2.3.2 Splicing	20
2.3.3 Inpainting	20
2.3.4 Resampling	21
2.4 Passive Detection of Digital Image Forgery	22
2.4.1 In-Camera-Based Detection	23
2.4.2 Post-Camera-Based Detection	31
2.5 Conclusion	34
3 Digital Image Authentication and Forgery Localization Using Demosaicing Artifacts	37
3.1 Introduction	37
3.1.1 Prior Art	37
3.1.2 Contributions and Organization	40
3.2 Statistical Analysis of Demosaicing Traces	41
3.2.1 Demosaicing Traces at Pixel Level	41
3.2.2 Demosaicing Artifacts in TIFF Images	42
3.2.3 Demosaicing Artifacts in JPEG Images	45

3.3	Feature Statistics for Demosaiced Images	46
3.3.1	CFA Pattern and Interpolation Kernel	47
3.3.2	Local Weighted Variance of Residues	47
3.3.3	Definition and Efficiency of Feature Statistics	48
3.4	Image Authentication and Forgery Localization	52
3.4.1	Statistical Model of Feature Statistic	52
3.4.2	Estimation of Model Parameters	52
3.4.3	Image Authentication and Forgery Localization	54
3.5	Numerical Experiments	56
3.5.1	Evaluation Criteria and Benchmark Algorithms	57
3.5.2	Authentication Performance	58
3.5.3	Localization Performance	59
3.6	Conclusion	63
4	Framework of JPEG DCT Coefficients Modeling and Forgery Localization	65
4.1	Introduction	65
4.1.1	Prior Art	65
4.1.2	Contributions and Organization	67
4.2	Recurrent Modeling Framework For JPEG DCT Coefficients	68
4.2.1	Authentic JPEG Images	68
4.2.2	Tampered JPEG Images	71
4.3	Framework Validation and Accuracy Assessment	73
4.3.1	Dataset Building	73
4.3.2	Framework Validation	74
4.3.3	Accuracy Assessment	77
4.4	Forgery Localization in Tampered JPEG Images	80
4.4.1	Forgery Localization Method	80
4.4.2	Performance Assessment	82
4.5	Conclusion	83
5	Statistical Detectors For Resampled TIFF Images	87
5.1	Introduction	87
5.2	Statistical Modeling of Resampled TIFF Images	88
5.2.1	Mathematical Analysis of Resampled Signals	88
5.2.2	Statistical Model for Resampled TIFF Pixels	92
5.3	Ideal Detector for Resampled TIFF Images	95
5.3.1	Analysis of TIFF Pixels Models	96
5.3.2	IID Data Extraction	97
5.3.3	LRT-Based Detector and Statistical Performance	99
5.4	Practical Detector for Resampled TIFF Images	103

5.4.1	Estimation of Model Parameters	104
5.4.2	GLRT-Based Detector and Performance	107
5.5	Numerical Experiments	108
5.5.1	Validation and Illustration	108
5.5.2	Performance Assessment	111
5.6	Conclusion	115
6	Conclusions and Perspectives	117
6.1	Conclusions	117
6.2	Perspectives	118
A	Appendix of Chapter 3	119
B	Appendix of Chapter 4	123
C	French Summary	127
C.1	Cadre de recherche	128
C.2	Plan de le thèse	130
C.3	Processus de traitement des images numériques	131
C.3.1	De la scène à l'image brute	131
C.3.2	De l'image brute à l'image TIFF	133
C.3.3	De l'image TIFF à l'image JPEG	135
C.4	Localisation des falsifications à partir du dématricage	136
C.5	Modélisation des coefficients DCT d'images JPEG et localisation des contrefaçons	140
C.6	Détecteurs statistiques pour les images TIFF rééchantillonnées	142
C.7	Conclusion	143
	References	145

Chapter 1

General Introduction

1.1 General Context

“*One picture is worth more than ten thousand words*”. The forcefulness of this old adage has been kept itself over the centuries until today. At the beginning, the *painting* was used to describe real-world happenings instead of *thousand words*. This role was assumed by the *photography* since the invention of still image cameras in the 19-th century. In this epoch, “*seeing is believing*” because images are captured by analog acquisition devices and stored on photograph films. The manipulation of images over such physical supports is extremely hard, time-consuming, costly and requires special skills of professionals through dark room tricks. Therefore, an image printed in a newspaper was commonly accepted as a certification of the truthfulness of the news. At the beginning of the 21-st century, the *digital revolution* has radically changed not only the way how an image is acquired, through digital devices such as digital cameras, smartphones, and tablets, etc., but also the way in which image contents are stored and transmitted. Laptops, USB keys, memory cards, compact discs are few examples of personal devices capable of storing digital images, while social networks (e.g., [Facebook](#), [Instagram](#), [Twitter](#)), cloud (e.g., [Google Drive](#), [Microsoft OneDrive](#), [iCloud](#)), websites (e.g., [Flickr](#), [Photobucket](#), [Google Photos](#)), are powerful services to store and share digital contents everywhere and anytime, through digital devices connected to the *Internet*. Thanks to such daily tools, almost everybody in our modern life is able to record, store and share a large number of digital images. Additionally, the dissemination of low-cost and powerful image editing software (e.g., [Adobe Photoshop](#), [GIMP](#), [Krita](#), [Pixlr](#)) have made the digital image alteration easier than ever before. This implies that images are not anymore something unchangeable, but rather like a *living organism* that evolves over time by means of various operations such as color modification, geometrical transformation, detail cancellation, content composition, etc. As a consequence, we have socially come to understand events in a far more visual way: “*there is more to an image than meets the eye*” [192], so that apparently “*seeing is no longer believing*”.

It does not matter if the image manipulation is merely for upright aims such as medical diagnosis, satire, beautifying, etc. Unfortunately, manipulated images have been abused for various malicious purposes related to all aspects of human life such as politics, ecology, race, gender, etc. [60]. In 1997,

58 innocent tourists were killed in a terrorist attack at the temple of Hatshepsut in Luxor Egypt (see Figure 1.1a). The *Swiss tabloid Blick* reported this event by modifying the color of the flooding water



(a) Original image



(b) Forged image

Figure 1.1: Event of terrorist attack at the temple of Hatshepsut in Luxor Egypt extracted from [39]

to appear as blood flowing from the temple (see Figure 1.1b), so that the public could understand the brutality of the attack. On July 09, 2008, numerous American news outlets, including the *Los Angeles Times*, the *Palm Beach Post*, and the *Chicago Tribune*, published a photo describing the Iranian missile test on their front-pages (see Figure 1.2a). After the publication, it was revealed that the second missile



(a) Photo appeared on newspapers



(b) Forged photo with duplication



(c) Genuine photo

Figure 1.2: Event of Iranian missile test

from the right was cloned in order to conceal a missile on the ground that did not fire (see Figure 1.2b). The genuine photo (see Figure 1.2c) appeared a day later further confirmed that the published picture was actually synthetic. In June 2010, the cover of *The Economist* showed President Barack Obama alone on a Louisiana beach examining the aftermath of the BP oil spill in the Gulf of Mexico (see Figure 1.3a). Under the headline “Obama v BP”, the president appears to be contemplating how to handle the political and ecological disaster. However, in July 2010, the *New York Times Media Decoder blog* revealed that the president was not alone (see Figure 1.3b). Some aforementioned events, among others¹, imply that forged images have become a big issue for society. In fact, they are appearing more and more frequently and sophisticatedly along with the growing advancement of new classes of artificial intelligence techniques (e.g., deepfakes [209, 212]) for which no specific technical know-how is required from the users. Such a serious situation leads to the need of efficient strategies and methods that allow to automatically verify the authenticity of digital images.

¹See e.g., <http://www.alteredimagesbdc.org/>

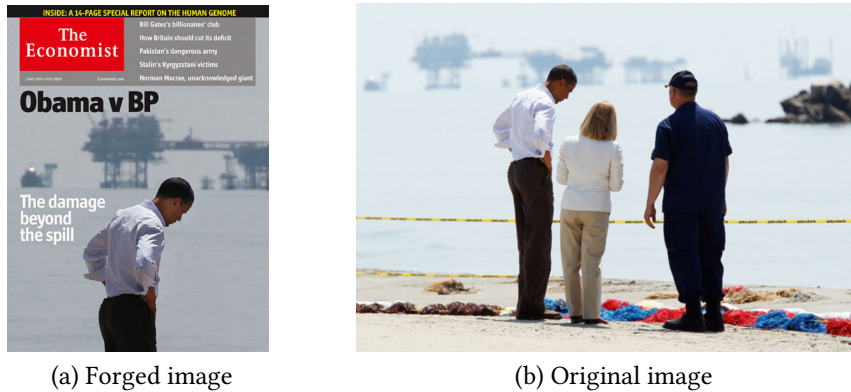


Figure 1.3: Event of President Barack Obama alone on a Louisiana beach

1.2 Research Framework

Digital image forensics have emerged as an indispensable research field to restore some trust to digital images. Generally, it is dedicated to seek answers for two major questions [171]:

1. *where does the digital image come from?*
2. *is the digital image still depicting the captured original scene?*

The first question is closely related to the problem of *image origin identification* with the specification of camera model or branch information and the determination of imaging mechanism as main aims [168]. The second question involves the problem of *image content integrity* where determining if an image is forged, which manipulation has been performed on the image, or which region of the image has been altered are fundamental tasks [114]. Despite their different interests, both the image origin identification and content integrity have mutual effects. Indeed, the knowledge about the device model or brand information can help forensic analysts know more about characteristics of acquisition devices, which leads to a potential improvement of detecting the underlying forgeries that could be performed in the inspected image. On the contrary, removing tampered area from a forged image allows to significantly narrow down the search range for the image origin. In their recent review [168], Pasquini *et al.* showed that the prevalence of existing digital image forensic works are mostly dedicated to the analysis of the acquisition source, either targeting the identification of the specific device or the camera mode. The contributions to forged image detection are however somewhat limited in contrast to their high spread on social media platforms. To meet this practical need, the present thesis focuses more especially on the development of methods for the image forgery detection and localization.

Actually, the literature of the forgery detection and localization has been addressed following two approaches [65]: (i) *active*, and (ii) *passive*. By active approach, some preset authentic information such as signature [70] and watermark [172] embedded in digital images is required to examine their truthfulness. However, such an embedded information is also the limitation of the approach, because it makes the production cost of digital cameras higher while its credibility remains questionable [208]. To remedy this drawback, the passive approach proposes to exploit intrinsic traces in the image acquisition or some specific traces left by forgeries to distinguish between tampered and natural images.

This approach is obviously suitable for most practical situations where only the suspect image is available in hand of forensic analysts and no authentic information is priorly known. So, it is not surprising that the passive approach has been increasingly developed in the past decade [20, 136]. More precisely, we can further classify the passive approach for forgery detection and localization into two main streams [212]: (i) *data-driven*, and (ii) *statistical model-based*. The first stream recommends to extract characteristic features from suspect images, and apply machine learning techniques to automatically learn discriminant features [12]. By taking account of various features in the learning, this approach allows to design universal tools which can deal with different kinds of malicious attacks [44]. Evidently, its performance and robustness depend heavily on the quality and the quantity of the considered data due to the data-driven nature. In the current state-of-the-art, this obstacle can be solved in part thanks to the deep learning, especially for the problems of cover source mismatch [170] and of dataset mismatch [161]. As an alternative, the second stream exploits incoherences in the statistical modeling of digital images to detect malicious attacks (see e.g., [63, 77]). It does not require a large amount of data as the first approach, but a deeper knowledge about the processing pipeline of digital cameras and malicious attacks. Very often, each developed model serves a specific goal, and hence is more adapted to dedicated forensic tools. The statistical modeling is also the main approach used in this doctoral project, however we target at versatile detectors which are applicable to different kinds of forgery techniques rather than to a certain specific manipulation.

1.3 Thesis Outline

The overall structure of the thesis is organized as follows.

- Chapter 1 introduces the general context of the research, the problem to be dealt with, as well as the employed methodology. The thesis outline and the author's publications are also presented in detail.
- Chapter 2 gives an overview on passive forgery detection based on the modeling of digital image processing pipeline. We start with an description of main image processing steps in a typical digital camera from image scene to Joint Photographic Experts Group (JPEG) image. Next, we introduce and illustrate common techniques of digital image forgery. Finally, we classify methods for image forgery detection following traces left by in-camera and post-camera processing steps.
- Chapter 3 focuses on the digital image authentication and forgery localization using demosaicing artifacts. The aim is to build an algorithm allowing a bridge between the color filter array pattern and demosaicing algorithm estimation, and the statistical analysis of demosaicing artifacts in spatial domain to improve the authentication and localization performance. After analyzing the evolution of demosaicing traces in camera acquisition pipeline, a robust feature statistic characterizing demosaiced digital images is first developed on the basis of the noise residue of green channel. Such a feature statistic is less sensitive to the edges problem because

only the smooth region of green channel is used in the development. Next, a single normal mixture model is proposed to describe the probability distribution of feature statistics for both original and tampered images. Therefore, normality tests can be used to authenticate automatically digital images. The authentication performance can be further improved by human interpretation of supported graphic tools. Finally, a penalized expectation-maximization (EM) algorithm is used to localize forged areas in tampered images. Many comparative studies on four well-known datasets show that the developed algorithm yields better performance and robustness than existing forensics algorithms of the same kind.

- Chapter 4 deals with the statistical modeling for discrete cosine transform (DCT) coefficients of both the authentic and tampered JPEG images and their application in forgery localization. In fact, various manipulations on JPEG images introduce single and multiple compression artifacts for forged and unmodified areas respectively. Based on the statistical analysis of compression cycle for authentic JPEG images and on the finite mixture paradigm, we propose a modeling framework for DCT coefficients of such tampered JPEG images. Its accuracy is numerically assessed using the Kullback-Leibler divergence on the basis of a dataset built from six well-known image databases. To illustrate the framework utility, an application in image forgery localization is proposed. By formulating the localization as a clustering problem, we use the plug-in Bayes rule combined with a simple EM algorithm to distinguish between forged and unmodified areas. Various experiments show that the proposed modeling framework yields high localization performances in terms of F_1 -score.
- Chapter 5 develops an efficient detector to distinguish between a resampled Tagged Image File Format (TIFF) image from an original TIFF image. To this end, we first propose a statistical model for resampled TIFF images by analyzing the complete processing process from a RAW image to a resampled TIFF image. Next, we formulate the detection problem as a likelihood ratio test between the models of original and re-sampled TIFF images. The test power is analytically evaluated in the context that all model parameters of original TIFF images are unknown. Numerous numerical experiments justify the performances of the detector.
- Chapter 6 concludes the major research finding and presents some perspectives of future works opened in the field of digital image forensics.
- Appendix A gives the formalization of the penalized EM algorithm used in Chapter 3. Appendix B represents the generalized Gamma distribution and the compound normal distribution used in Chapter 4, as well as their parameters estimation. A short French summary of this thesis is also provided in Appendix C.

1.4 Publications

Most of the material presented in this thesis appears in the following publications that represent original work, of which the author has been the main contributor.

Journal Article

- Le Nhan, and Florent Reira. An improved algorithm for digital image authentication and forgery localization using demosaicing artifacts. *IEEE Access* 7: 125038-125053, 2019.

Journal Articles Under Review

- Le Nhan, and Florent Reira. A Recurrent Framework for Statistical Modeling of JPEG AC DCT Coefficients. Submitted to *IEEE Access*, 2021.
- Le Nhan, and Florent Reira. Statistical Modeling Framework For AC DCT Coefficients of Tampered JPEG Images and Application in Forgery Localization. Submitted to *IEEE Transactions on Information Forensics and Security*, 2021.

Conference Paper

- Le Nhan, and Florent Reira. Statistical Detector of Resampled TIFF Images. In *2018 IEEE International Symposium on Signal Processing and Information Technology*, pages 398-401, 2018.

Chapter 2

Overview on Digital Image Processing Pipeline and Passive Forgery Detection

2.1 Introduction

As stated in Chapter 1, this thesis relies mainly on the statistical model-based approach to detect forged digital images and to localize their tampered regions. Such an approach requires a good knowledge about the image processing pipeline of a digital camera, as well as about the forgery techniques. This is why this chapter starts with an introduction of digital camera image formation in Section 2.2. For an easier comprehension, we divide the image formation pipeline into three parts according to possible image storage formats generated by the camera. For each part, we analyze the main processing steps therein, and present the key statistical models of the output image. Next, we describe and illustrate four commonly used techniques of digital image forgery in Section 2.3: (i) *cloning*, (ii) *splicing*, (iii) *in-painting*, and (iv) *resampling*. Subsequently, we provide in Section 2.4 an overview of various passive methods for image forgery detection exploiting inconsistencies existing in the digital image processing pipeline. We category these methods following the traces left by in-camera and post-camera processing steps. Finally, Section 2.5 concludes the chapter and specifies more concretely the problems to be dealt with in this thesis.

2.2 Image Processing Pipeline in Digital Still Cameras

Forgery analysis of digital images requires a knowledge about the creation of those images. This section provides increased insight into digital camera image formation. Although several devices can be used for digital imaging such as scanner, graphic tablet, etc., we just limit our study to digital still cameras¹ because they have gained significant popularity in recent years. The whole processing pipeline of such a camera from light capturing to image storage is illustrated by Figure 2.1. In the following review, we divide this processing pipeline into three parts according to possible image storage formats

¹In the following, expect explicitly stated to avoid any misunderstanding, *digital still camera* is simply called *camera* or *digital camera* for short.

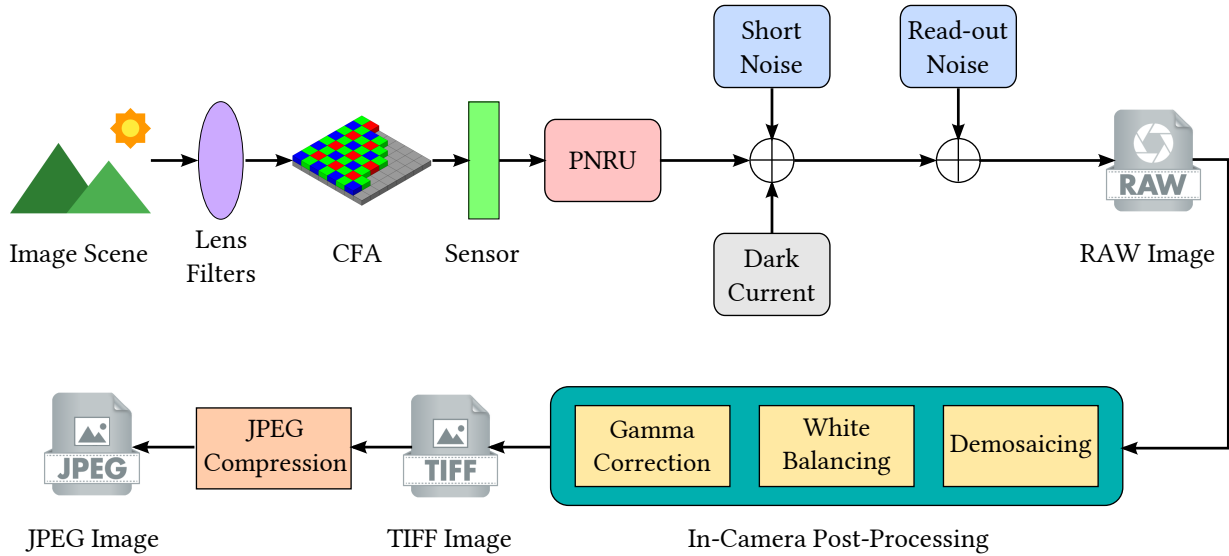


Figure 2.1: Illustration of a typical acquisition pipeline in a digital still camera

generated by the camera: (i) from image scene to RAW image, (ii) from RAW image to TIFF image, and (iii) from TIFF image to JPEG image. For each part, we analyze the main processing steps therein, and introduce the key statistical models of the output image.

2.2.1 From Image Scene to RAW Image

Digital still cameras imitate the human visual system. They consist of a lens module, sampling filters, color filter array (CFA), imaging sensor, and a digital image processor [1]. The lens module is essentially composed of a lens and the mechanisms to control exposure, focusing, and image stabilization to collect and pilot the light coming from the real scene. After the light enters the camera via the lens, it goes through a combination of filters that includes at least the infra-red and anti-aliasing filters to ensure maximum visible quality. The light is next focused onto imaging sensor, an array of rows and columns of light-sensing elements called pixels. Two common types of an image sensor are charge-coupled device (CCD) and complementary metal-oxide semiconductor (CMOS). Each light sensing element of sensor array integrates the incident light over the whole spectrum and obtains an electric signal representation of the scenery. Since each imaging sensor element is essentially monochromatic, an easy way to capture color images is to separate sensor chips for each color component. However, such a low-tech and costly solution does not allow a competitive advantage for enterprises. This is why most digital camera devices in the market only use a single CCD or CMOS sensor chip coupled with a CFA instead [142]. The CFA arranges pixels in a pattern so that each element has a different spectral filter. Hence, each element only senses one band of wavelength, and the RAW image collected from the imaging sensor is a mosaic of different colors and varying intensity values.

Among many CFA patterns with different primary colors (see e.g., [8]), the one designed by Bayer [14] is most used in commercial digital cameras. It measures the green component on a *quincunx* grid, the red and blue components on *rectangular* grids. The higher sampling rate for the green channel allows a better capture of the light luminance, and therefore provides better image quality [84]. The

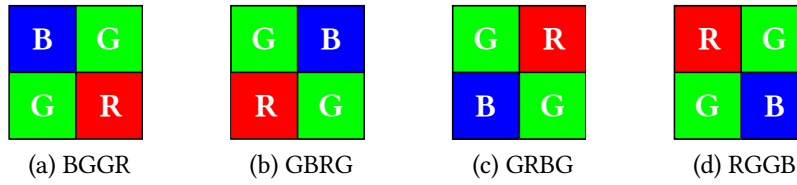


Figure 2.2: Four possible configurations of the Bayer pattern

Bayer pattern has itself four possible configurations as depicted in Figure 2.2. We seek now a mathematical representation \mathbf{V} of a RAW image captured along with a configuration \mathbf{B} of Bayer pattern. As illustrated in Figure 2.3, the RAW image \mathbf{V} is of single-channel because the Bayer configuration \mathbf{B} allows each pixel to record only one color channel c among the red (r), the green (g) or the blue (b). Therefore, we can express \mathbf{V} as a two-dimensional matrix of size $M \times N$ whose pixel value at the

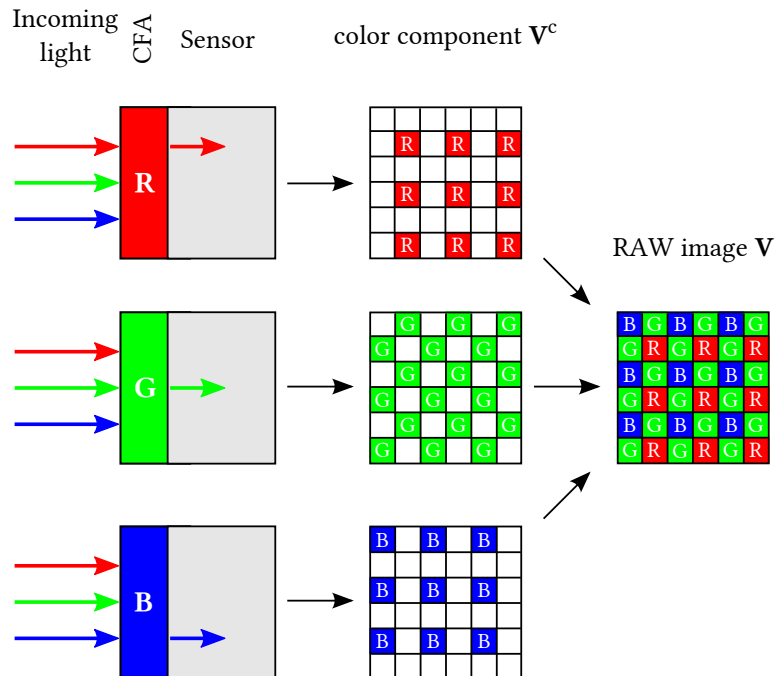


Figure 2.3: Formulation process of the RAW image

location (m, n) of the color component \mathbf{V}^c is given by

$$v_{m,n}^c = \begin{cases} v_{m,n} & \text{if } \mathbf{B}_{m,n} = c, \\ 0 & \text{otherwise,} \end{cases} \quad (2.1)$$

where $1 \leq m \leq M$, $1 \leq n \leq N$, and $\mathbf{B}_{m,n}$ denotes the location (m, n) of the Bayer configuration.

The RAW image acquisition process is disturbed by several unavoidable noise sources whose shot noise, dark current noise, read-out noise and photo-response non-uniformity (PRNU) noise are dominant ones (see Figure 2.1). The shot noise, also known as Poisson-distributed noise, has its origin in the quantum nature of light. The dark current noise, also referred as fixed pattern noise, is generated

by the thermal energy in the light absence. The read-out noise encompasses all electronic noises involved in the acquisition chain. Finally, the PRNU noise accounts for differences of pixels response to the incident light due to the imperfections during the sensor manufacturing process (e.g., the heterogeneity of silicon wafers). Considering these noise sources, we can further express the pixel $v_{m,n}$ in (2.1) as

$$v_{m,n} = \mu_{v_{m,n}} + \epsilon_{v_{m,n}}, \quad (2.2)$$

where $\mu_{v_{m,n}}$ denotes the expectation of $v_{m,n}$ in the absence of noise, and $\epsilon_{v_{m,n}}$ stands for all interfered noise sources. While the expectation $\mu_{v_{m,n}}$ is merely seen as deterministic signal and not be modeled², the noise $\epsilon_{v_{m,n}}$ should be carefully studied. Indeed, the noise models for RAW images can be categorized into two classes in the literature: (i) *signal-independent*, and (ii) *signal-dependent*. If the former assumes the stationarity of noise in the whole image, the latter takes into account the proportional dependence of noise variance on the original pixel intensity. The well-known additive white Gaussian noise model (see e.g., [22]) is a typical example of the signal-independent noise class. Meanwhile, signal-dependent noise class includes Poisson or film-grain noise model [119], Poisson-Gaussian noise model [141, 149], heteroscedastic noise model [67, 205], non-linear noise model [58], etc. Compared to the first class, the second one is more accurate, and obviously more difficult to be dealt with.

2.2.2 From RAW Image to TIFF Image

To render a full-color uncompressed TIFF image from the RAW image, several post-processing operations are carried out (see e.g., [53]), where *demosaicing*, *white balancing* and *gamma correction* are the significant ones.

The demosaicing, also known as *CFA interpolation*, allows to fill up the missing pixels due to the CFA sampling by using their neighborhoods [185]. Generally, we can classify algorithms for demosaicing into two classes: (i) *non-adaptive*, and (ii) *adaptive*. Non-adaptive algorithms use the same interpolation technique for all pixels, so that in most cases, they can be simply defined by a single *interpolation kernel* \mathbf{K}^c for each color channel c (see also Table 2.1). Consequently, the color component \mathbf{X}_D^c of the demosaiced image \mathbf{X}_D can be computed from the RAW component \mathbf{V}^c as

$$\mathbf{X}_D^c = \mathbf{K}^c * \mathbf{V}^c, \quad (2.3)$$

where $*$ denotes the two-dimensional convolution operation. The non-adaptive demosaicing algorithms provide satisfactory results in smooth regions, but usually fail in textured regions and edges. Adaptive algorithms can solve this problem. They take into account edge information, as well as inter-channel correlation to find an appropriate set of coefficients which permits to minimize the overall interpolation error (see e.g., [42, 210]). This makes the adaptive algorithms more accurate than the non-adaptive ones at a higher price of computationally intensive.

After the demosaicing, we need the white balancing to remove unrealistic color casts [186]. In fact, due to the color temperature difference of light sources, a shift of the reflection spectrum of the

²The expectations differ from each other due to heterogeneity in a natural RAW image.

name	red and blue channels	green channel
bi-linear	$\mathbf{K}^r = \mathbf{K}^b = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	$\mathbf{K}^g = \frac{1}{4} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$
bi-cubic	$\mathbf{K}^r = \mathbf{K}^b = \frac{1}{256} \begin{bmatrix} 1 & 0 & -9 & -16 & -9 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -9 & 0 & 81 & 144 & 81 & 0 & -9 \\ -16 & 0 & 144 & 256 & 144 & 0 & -16 \\ -6 & 0 & 81 & 144 & 81 & 0 & -6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -9 & -16 & -9 & 0 & 1 \end{bmatrix}$	$\mathbf{K}^g = \frac{1}{256} \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -9 & 0 & -9 & 0 & 0 \\ 0 & -9 & 0 & 81 & 0 & -9 & 0 \\ 1 & 0 & 81 & 256 & 81 & 0 & 1 \\ 0 & -9 & 0 & 81 & 0 & -9 & 0 \\ 0 & 0 & -9 & 0 & -9 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$

Table 2.1: Examples of interpolation kernel used in non-adaptive demosaicing algorithms

object from its true color may occur. This shift makes the object appear different in color when it is illuminated under different light sources. The white balance adjustment is therefore implemented in digital cameras to compensate this illumination imbalance. Basically, it is performed by multiplying pixels in each color channel \mathbf{X}_D^c by a different gain factor g_W^c to render a white-balanced image \mathbf{X}_W

$$\mathbf{X}_W^c = g_W^c \cdot \mathbf{X}_D^c, \quad (2.4)$$

where \mathbf{X}_W^c denotes the component of \mathbf{X}_W associated with the color channel c . The prior knowledge of light sources is critical to select appropriate gain factors g_W^c . In the absence of this knowledge, g_W^c can be estimated by several algorithms described in [121]. For an example, we just introduce hereinafter a simple algorithm based on the *gray world* assumption: *the average reflectance of a scene is achromatic*. This implies that the average values of three color channels in a given scene are roughly equal, so that

$$\bar{x}_W^r = \bar{x}_W^g = \bar{x}_W^b, \quad (2.5)$$

where \bar{x}_W^c , with $c \in \{r, g, b\}$, denotes the average intensity of \mathbf{X}_W^c computed by

$$\bar{x}_W^c = \frac{1}{M \cdot N} \cdot \sum_{m=1}^M \sum_{n=1}^N x_{W,m,n}^c. \quad (2.6)$$

Using (2.4) and (2.5), it follows that

$$g_W^r \cdot \bar{x}_D^r = g_W^g \cdot \bar{x}_D^g = g_W^b \cdot \bar{x}_D^b, \quad (2.7)$$

where \bar{x}_D^c denotes the average intensity of \mathbf{X}_D^c computed in the same way as (2.6). Since the human eye is more sensitive to the green color, the algorithm proposes to keep the green channel as it is,

hence

$$g_W^b = 1. \quad (2.8)$$

Therefore, the gain factors of the other color channels are derived from (2.7) as

$$g_W^r = \frac{\bar{x}_D^g}{\bar{x}_D^r} \quad \text{and} \quad g_W^b = \frac{\bar{x}_D^g}{\bar{x}_D^b}. \quad (2.9)$$

Introducing (2.8) and (2.9) into (2.4), we finally obtain the white-balanced image X_W .

At this stage, the pixel intensity only appears the linear characteristic with respect to the RAW image intensity [53]. This linearity does not fit in with most display devices whose transfer function represents a power relationship between the luminance L and the voltage U

$$L = V^\gamma, \quad (2.10)$$

where γ is a constant specific for each device. Especially, $\gamma = 2.2$ for the cathode ray tube monitor. The gamma correction is thus necessary to compensate this effect and render the luminance into a perceptually uniform domain. It is simply the inverse of (2.10) applying to each pixel value of X_W such that

$$x_{m,n}^c = \left(x_{W,m,n}^c\right)^{\frac{1}{\gamma}}, \quad (2.11)$$

where $x_{m,n}^c$ denotes the pixel at location (m, n) of the component X^c of the TIFF image.

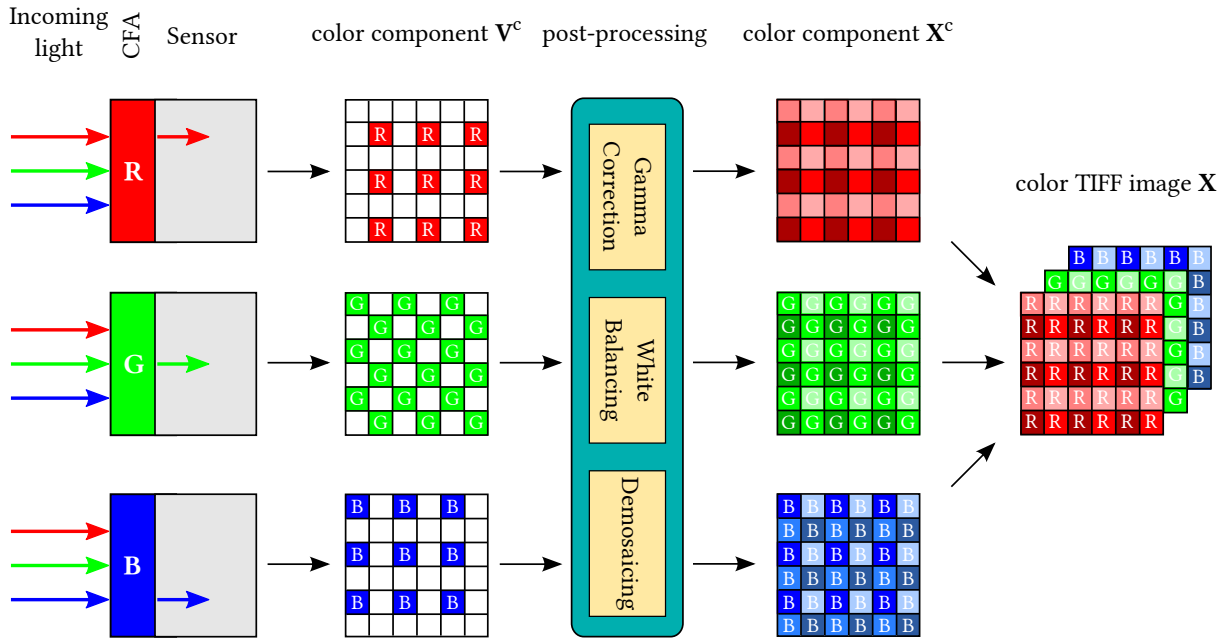


Figure 2.4: Formulation process of the TIFF image

At the output of the post-processing process, we obtain an uncompressed full-color TIFF image X characterized by a three-dimensional matrix of size $M \times N \times 3$ (see Figure 2.4). To statistically describe the TIFF image in the spatial domain, both the *black-box* and *white-box* approaches can be resorted to.

By the black-box approach, we attempt to establish a so-called *camera response function* $f_{CRF}(\cdot)$ that maps image irradiance³ to TIFF image intensity

$$x_{m,n} = f_{CRF}(\mu_{v_{m,n}} + \epsilon_{v_{m,n}}), \quad (2.12)$$

where $\mu_{v_{m,n}}$ denotes the image irradiance and $\epsilon_{v_{m,n}}$ accounts for all noise sources. Gamma curve [59], polynomial curve [82] and generalized Gamma curve [159] are some parametric models for $f_{CRF}(\cdot)$. The main drawback of such an approach is that the study of noise statistics is relatively hard. A potential solution is to linearize $f_{CRF}(\cdot)$ using the first order of Taylor's series expansion [58]

$$x_{m,n} = f_{CRF}(\mu_{v_{m,n}} + \epsilon_{v_{m,n}}) \simeq f_{CRF}(\mu_{v_{m,n}}) + f'_{CRF}(\mu_{v_{m,n}}) \cdot \epsilon_{v_{m,n}}, \quad (2.13)$$

where $f'_{CRF}(\cdot)$ is the first derivative of $f_{CRF}(\cdot)$. Therefore, the noise $\epsilon_{x_{m,n}}$ of $x_{m,n}$ can be expressed in function of $\epsilon_{v_{m,n}}$ as follows

$$\epsilon_{x_{m,n}} = f'_{CRF}(\mu_{v_{m,n}}) \cdot \epsilon_{v_{m,n}}. \quad (2.14)$$

Unlike the black-box approach, the white-box approach requires a deeper understanding of the post-processing process for the step-by-step modeling. A typical example is the way to develop the so-called *generalized signal-dependent noise* model in [207]. Indeed, starting from the *heteroscedastic noise* model of RAW pixels proposed in [67], Thai *et al.* have followed the effects of demosaicing, white-balancing and gamma correction to build their noise model for TIFF images. Even not all steps in the post-processing process are taken into account, numerical results therein show clearly that a better insight of the in-camera post-processing provides more accurate models than the black-box approach. Some extensions and applications of the generalized signal-dependent noise model can be found in [179, 208]. Besides the modeling in spatial domain, we can also exploit the discrete cosine transformation (DCT) domain to characterize the TIFF image. In fact, all statistical models of primary unquantized DCT coefficients can be employed for this goal. Subsection 4.1.1 of Chapter 4 gives a review of these models.

2.2.3 From TIFF Image to JPEG Image

Due to its high quality, the TIFF image is not really adapted for the storage or transmission, especially in large numbers. This is why most digital cameras employ a *lossy* compression algorithm to reduce the image data size. Such an algorithm attempts to eliminate the image information that is not visually significant, hence it is irreversible in the sense that the image reconstructed from the compressed data is no longer identical to the original TIFF image. Among many lossy compression algorithms (see e.g., [92] for a recent survey), the Joint Photographic Experts Group (JPEG) standard [213] is most popular thanks to the good compromise between image visual quality and size. This motivates us to focus our study on the JPEG compression. As depicted in Figure 2.5, a typical JPEG compression chain consists of four fundamental steps: (i) *color transformation*, (ii) *discrete cosine transformation* (iii), *quantization*,

³Light energy incident on image sensors

and (iv) *entropy coding*.

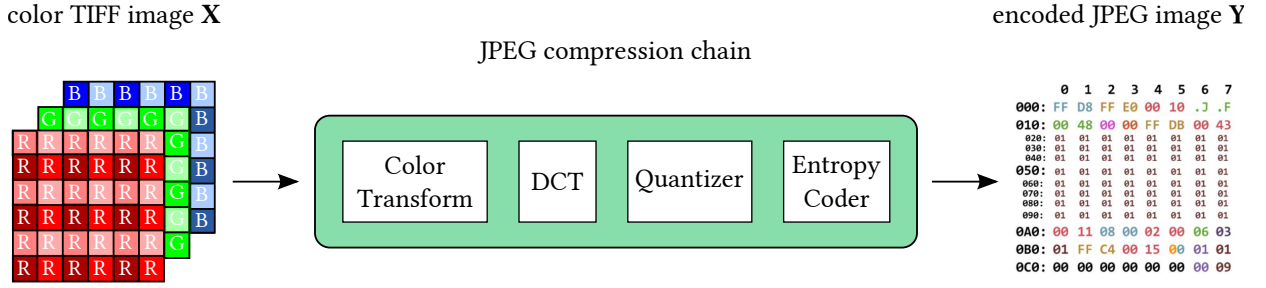


Figure 2.5: Key steps in JPEG compression chain

Since the JPEG compression works better under YCbCr color space [53], a transformation of TIFF image from the RGB color space to the YCbCr color space is first carried out

$$\begin{bmatrix} \tilde{X}^y \\ \tilde{X}^{cb} \\ \tilde{X}^{cr} \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & 0.081 \end{bmatrix} \cdot \begin{bmatrix} X^r \\ X^g \\ X^b \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}, \quad (2.15)$$

where X^r , X^g and X^b are the three components of the TIFF image X in the RGB color space, and \tilde{X}^y , \tilde{X}^{cb} and \tilde{X}^{cr} are the three components of the transformed TIFF image \tilde{X} in the YCbCr color space. Since this transformation is simply a linear operation, it is surely lossless.

Next, for each component \tilde{X}^y , \tilde{X}^{cb} or \tilde{X}^{cr} , we separately apply the DCT to their 8×8 blocks to convert pixel values from spatial domain into transform coefficients

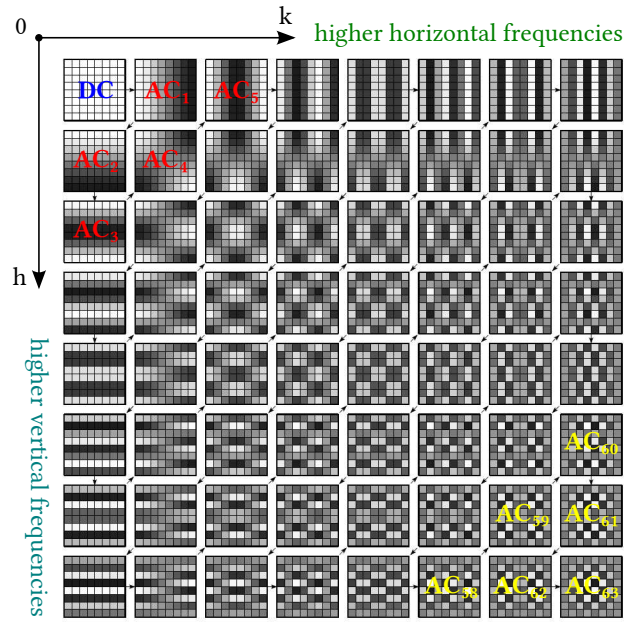
$$C_{h,k} = \frac{1}{4} \cdot T_h \cdot T_k \cdot \sum_{m=0}^7 \sum_{n=0}^7 \tilde{x}_{m,n} \cdot \cos\left(\frac{(2m+1)h\pi}{16}\right) \cdot \cos\left(\frac{(2n+1)k\pi}{16}\right), \quad (2.16)$$

where $\tilde{x}_{m,n}$, $0 \leq m, n \leq 7$, denotes a pixel in a 8×8 block of \tilde{X}^4 , T_h and T_k are the normalized weight given by

$$T_h = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } h = 0, \\ 1 & \text{if } h > 0. \end{cases} \quad (2.17)$$

Consequently, we obtain 64 two-dimensional DCT coefficients $C_{h,k}$, $0 \leq h, k \leq 7$, for each 8×8 pixel block. These coefficients are arranged in a 8×8 block such that the low-frequency elements are located at the upper left corner, while the high-frequency elements distribute in the lower right corner (see Figure 2.6). The coefficient $C_{0,0}$, representing the mean value of pixels in the 8×8 block, is referred as *direct current* (DC) because it is analogous to a zero frequency, and 63 others are called *alternative current* (AC). Conveniently, for natural images, the majority of energy concentrate in the DC and low AC frequency bands, while very detail information is associated with high AC frequencies. Therefore, we can discard the higher AC frequency bands to reduce the data size without much impact on the

⁴To simplify, the indices y , cb and cr are omitted here as each component of the transformed TIFF image is processed separately.

Figure 2.6: The DCT basis in 2D for 8×8 pixels block

visual quality. Moreover, the DCT is especially good at compacting most of the information into a small number of coefficients when neighboring pixel values are highly correlated. This makes it an especially good transform to use for compressing images of interest to human beings. Mathematically, the DCT is merely a change of basis, so it is a lossless operation.

The elimination of high-frequency AC DCT coefficients is done through the quantization step. It proposes to divide each DCT coefficient $C_{h,k}$ by the corresponding element $q_{h,k}$ of a quantization table T , then round up the result to the nearest integer to obtain the *quantized* DCT coefficient

$$D_{h,k} = \left\lfloor \frac{C_{h,k}}{q_{h,k}} \right\rfloor, \quad 0 \leq h, k \leq 7, \quad (2.18)$$

where $\lfloor \cdot \rfloor$ denotes the nearest integer rounding. The quantization table T is designed with regard to the human visual system. Since the sensitivity of human eyes to luminance and chrominance information is different, it is recommended to use one quantization table for the luminance component, and another quantization table for the chrominance components. Such tables, indexed by a quality factor $QF \in \{1, 2, \dots, 100\}$, are computed as follows

$$T_{QF} = \begin{cases} \max \left\{ 1, \left\lfloor 2 \cdot T_{50} \cdot \left(1 - \frac{QF}{100} \right) \right\rfloor \right\} & \text{if } QF > 50, \\ \min \left\{ 255 \cdot 1, \left\lfloor T_{50} \cdot \frac{50}{QF} \right\rfloor \right\} & \text{if } QF \leq 50, \end{cases} \quad (2.19)$$

where 1 denotes the 8×8 matrix of ones, T_{50} is given in Table 2.2. Larger values in the quantization table correspond to coarser quantization, and hence greater compression. Figure 2.7 illustrates the JPEG compression process of an 8×8 luminance block up to the quantization step with T_{50} . Clearly, the quantization is a lossy operation.

luminance channel								chrominance channels							
$T_{50} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$								$T_{50} = \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}$							

Table 2.2: 50% quality standard JPEG quantization tables

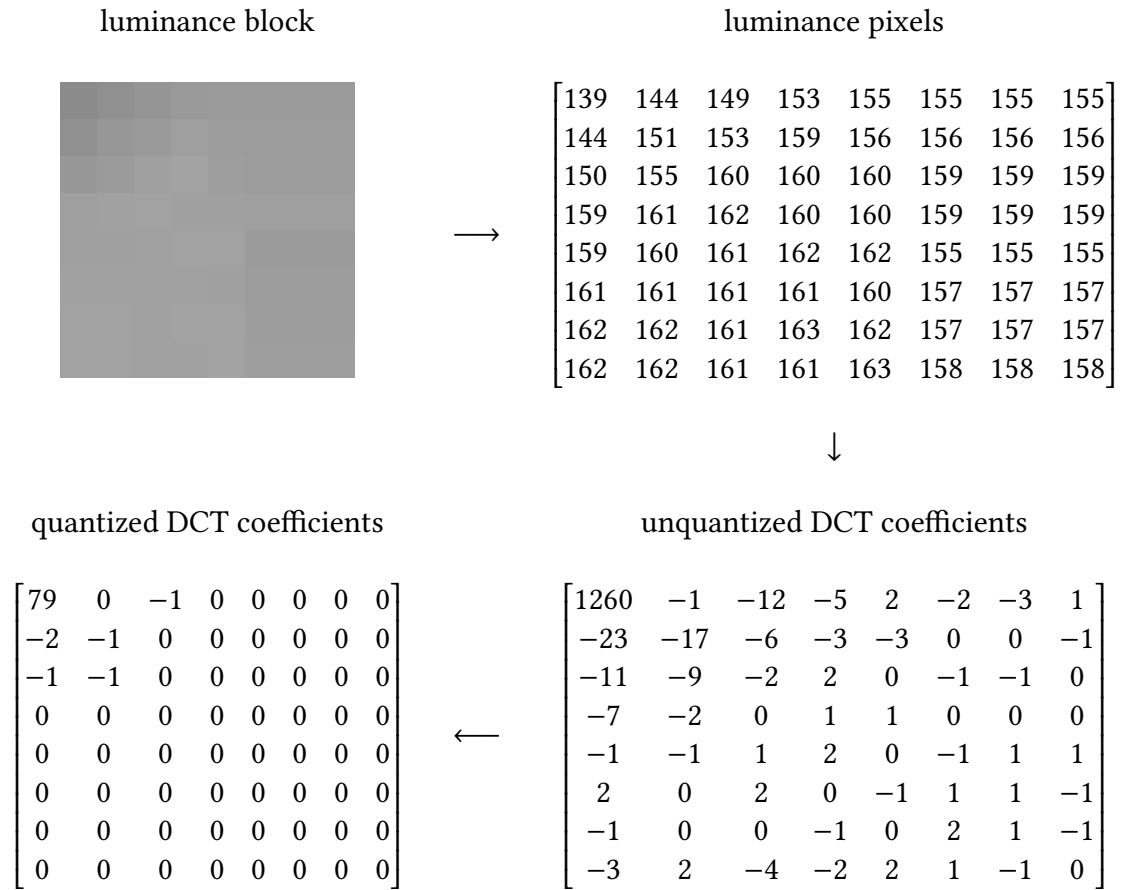


Figure 2.7: An 8×8 luminance block, its unquantized and quantized DCT coefficients with T_{50}

Finally, the quantized DCT coefficients are fed to a lossless entropy coder whose mechanism is a combination of *Huffman* coding and *run-length* encoding principles. Due to the high correlation between adjacent blocks in natural images, the JPEG standard recommends to encode the difference of inter-block DC coefficients by a Huffman code instead of their value directly. Meanwhile, a run-length encoding algorithm is applied for intra-block AC coefficients in zig-zag order (see Figure 2.6)

to take advantage of long sequences of zero-values inherent in quantized DCT blocks. The output of the entropy coder is an encoded JPEG image Y in bitstream (see Figure 2.5).

Because of bitstream form, JPEG images cannot be displayed on a monitor device directly, but through a decompression chain (see Figure 2.8). Basically, it is the converse process of the JPEG

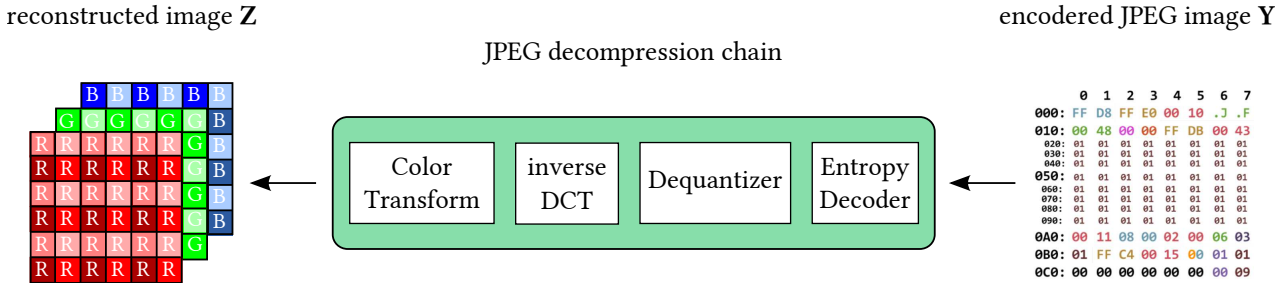


Figure 2.8: Key steps in JPEG decompression chain

compression chain: (i) *entropy decoding*, (ii) *dequantization*, (iii) *inverse discrete cosine transformation*, and (iv) *color inverse transformation*. A summary of this process is briefly introduced here. Firstly, a lossless entropy decoding is applied to JPEG bitstream Y to perfectly return the two-dimensional quantized DCT coefficients $D_{h,k}$. Subsequently, $D_{h,k}$ is multiplied by the same quantization step $q_{h,k}$ as in the compression chain to generate two-dimensional *dequantized* DCT coefficients $\widehat{C}_{h,k}$

$$\widehat{C}_{h,k} = q_{h,k} \cdot D_{h,k}. \quad (2.20)$$

Next, the inverse DCT is applied to $\widehat{C}_{h,k}$ to render image in YCbCr color space

$$\hat{x}_{m,n} = \sum_{h=0}^7 \sum_{k=0}^7 \frac{1}{4} \cdot T_h \cdot T_k \cdot \widehat{C}_{h,k} \cdot \cos\left(\frac{(2m+1)h\pi}{16}\right) \cdot \cos\left(\frac{(2n+1)k\pi}{16}\right), \quad (2.21)$$

where T_h and T_k are the normalized weight given by (2.17). Finally, a color transformation of $\hat{x}_{m,n}$ is performed to obtain the corresponding pixel value $\hat{z}_{m,n}$ in the RGB color space. Since this value is possibly not integers or beyond a finite dynamic range (e.g., [0, 255] for 8-bits images), the nearest integer rounding⁵ and truncation operation is required to finalize the decompression process

$$z_{m,n} = \text{trunc}(\lceil \hat{z}_{m,n} \rceil), \quad (2.22)$$

where $\text{trunc}(\cdot)$ denotes the truncation operation. In general, the reconstructed image Z differs from its original TIFF image X because of the quantization, rounding and truncation errors. For an illustration, we present in Figure 2.9 the inverse process of the chain in Figure 2.7.

Now, we show how to characterize JPEG images. Basically, all approaches used to model TIFF images can serve this purpose: *black-box* approach in the spatial domain (see e.g., [90, 137, 214]), *white-box* approach in the spatial domain (see e.g., [179, 189]), and DCT coefficients-based approach

⁵Although the nearest integer rounding is represented here, other kinds of rounding like floor or ceiling are possible. In fact, the different choices of rounding operation lead to JPEG dimple artifacts mentioned later in Subsection 2.4.1.4.

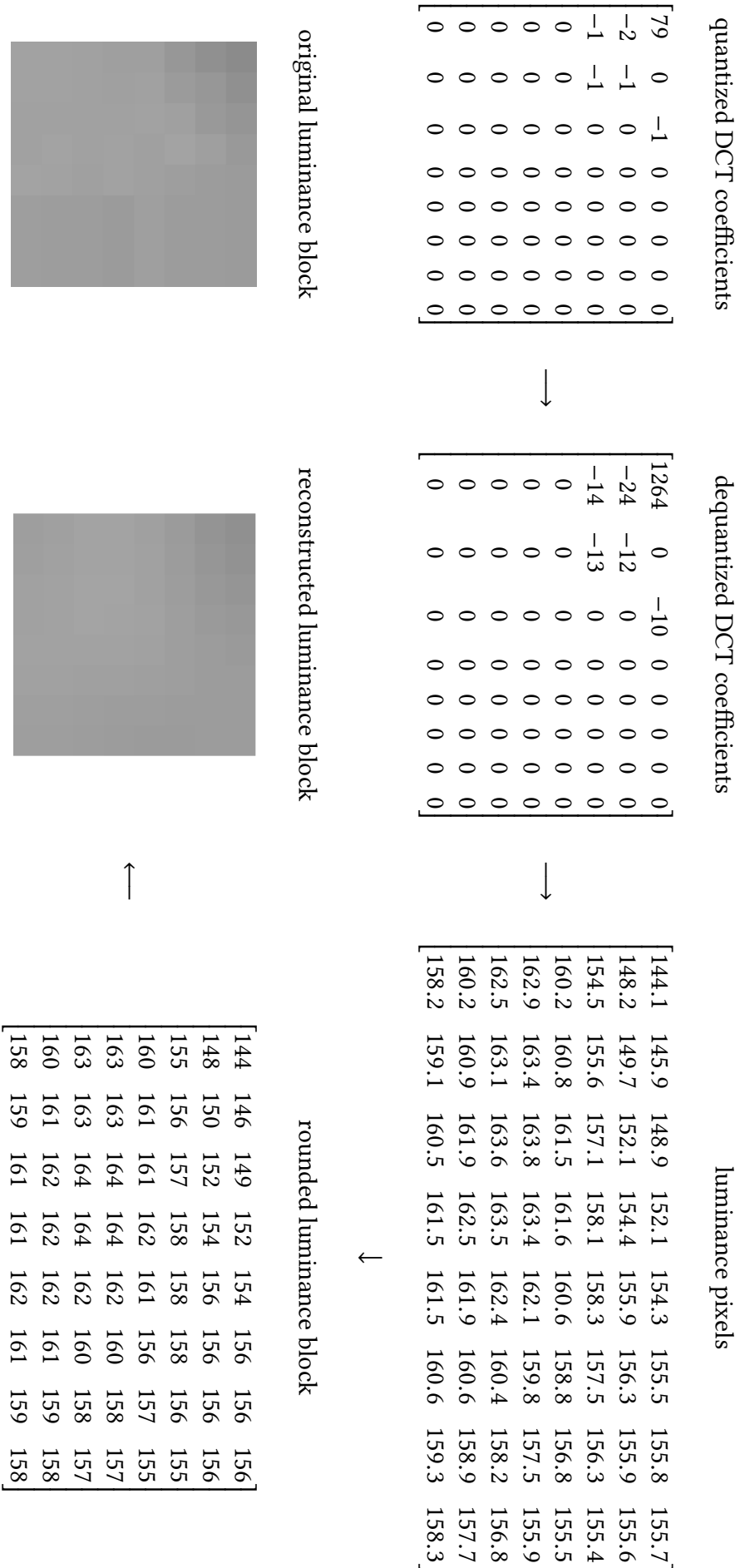


Figure 2.9: Inverse process of the transformation chain in Figure (2.7)

(see e.g., [132, 167, 206]). We are interested more specially in models developed by the last approach because the DCT is the basis of JPEG compression. We can classify them into two categories: (i) *first-order statistics*, and (ii) *higher-order statistics*. The models in the first category consider that the DCT coefficients at the same frequency are independent identically distributed (IID) realizations of a random variable. Therefore, many statistical distributions can be used to fit in with the DCT coefficients. Most of them are empirical models conducted by goodness-of-fit tests on a set of images (see e.g., [29, 157, 188]). The first mathematical analysis for DCT coefficients is given by Lam and Goodman in [122]. However, such an analysis had not yet been finalized until Thai *et al.* successfully studied the block variance model of DCT coefficients and approximated it by an Gamma distribution in [206]. Notwithstanding, the IID assumption made by the first-order models is not always true because of the correlation inherent in the DCT coefficients of natural images. The higher-order models can solve this problem by considering two fundamental kinds of correlation [211]: (i) *intra-block*, and (ii) *inter-block*. The intra-block correlation reflects the dependence between adjacent DCT coefficients within a same 8×8 block, while the inter-block correlation take into account the dependence of DCT coefficients at the same frequency between neighboring 8×8 blocks. Both the correlations can be expressed and measured by co-occurrence matrices and probability transfer matrices in the direction of horizontal, vertical, main and secondary diagonals (see e.g., [85, 113, 133]).

2.3 Techniques of Digital Image Forgery

We provide in this section an overview of forgery techniques for digital images. Although many forgeries can be introduced throughout the digital image processing pipeline (see e.g., [11, 61, 201]), we just limit our study to very common post-camera techniques including: (i) *cloning*, (ii) *splicing*, (iii) *inpainting*, and (iv) *resampling*. For each of techniques, we state the definition and its role in digital image forgery.

2.3.1 Cloning

Cloning, also known as *copy-move* forgery, is perhaps one of the easiest, yet powerful, techniques to create forged images. It consists of copying areas of an image, then pasting them elsewhere within the same image. The objective is to hide (see Figure 2.10) or duplicate (see Figure 2.11) some areas

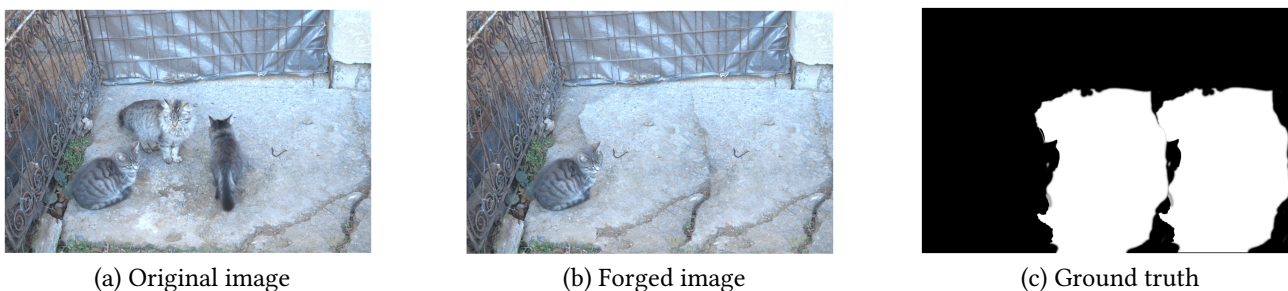


Figure 2.10: Cloning with covering attack taken from the Image Manipulation Dataset [41]

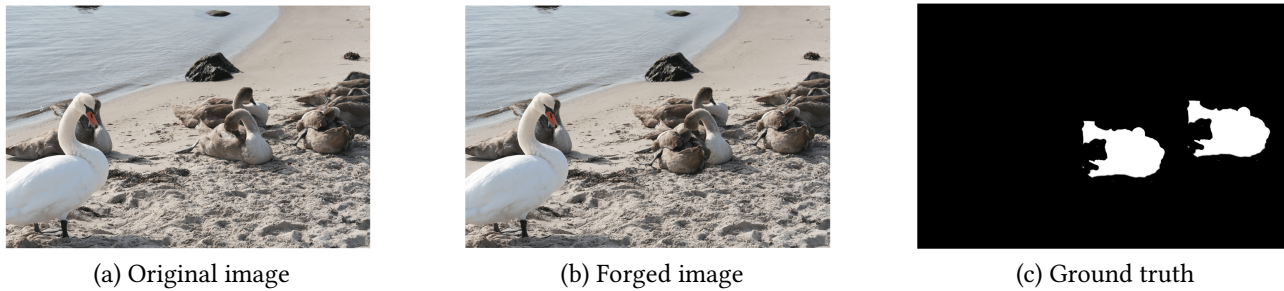


Figure 2.11: Cloning with duplication attack given from the Image Manipulation Dataset [41]

of the image. Since both the source and target areas stem from the same image, properties like the color temperature, illumination conditions and noise are expected to be well-matched between the tampered areas and the image. This is why the cloning detection is relatively challenging, especially when the manipulation only involves small or smooth regions, or when the forged areas have been additionally processed by some severe attacks, such as large-scale resizing and heavy noise addition.

2.3.2 Splicing

Splicing, also referred as *cut-paste* forgery, is another common technique of digital image forgeries. It attempts to alter the content and meaning of an original image by replacing some areas in the image by those from others using cutting and pasting operations (see Figure 2.12 for an example). Com-

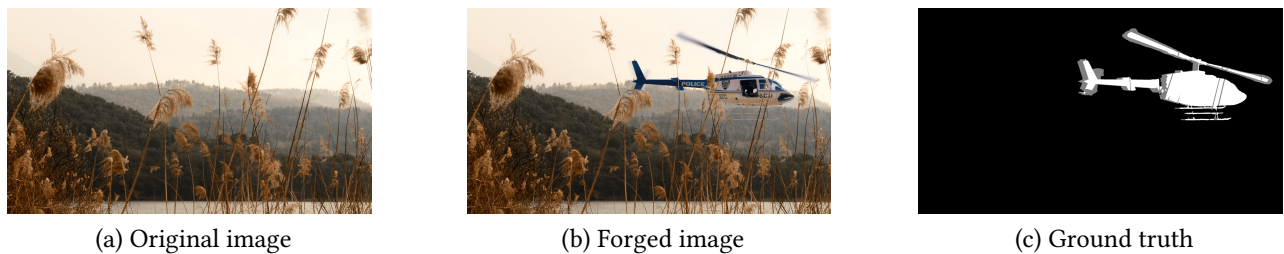


Figure 2.12: Splicing attack taken from the Realistic Tampering Dataset [115]

pared to the cloning, the splicing is far more flexible thanks to different image sources, and therefore allows forged images with a very different content than the original. However, the multiple image sources also induce more inconsistencies in image statistical properties, especially at the borderlines. Of course, when performed carefully, the border between the spliced regions can be still visually imperceptible. If necessary, the splicing may be accompanied with other operation such as rotation, resizing, compression, noise addition, filtering, and contrast and brightness adjustments to make the final forgeries more convincing.

2.3.3 Inpainting

Inpainting is originally proposed to repair old photos from the early days (see Figure 2.13). In image tampering, it is also called *erase-fill* forgery [232] because the main attacks is to erase some undesired



Figure 2.13: Inpainting for restoration purpose taken from [17]

object in an image and fill the resulted hole with neighboring texture (see Figure 2.14 for an example). Traditionally, we can classify inpainting methods into two categories [83]: (i) *diffusion-based*, and (ii)

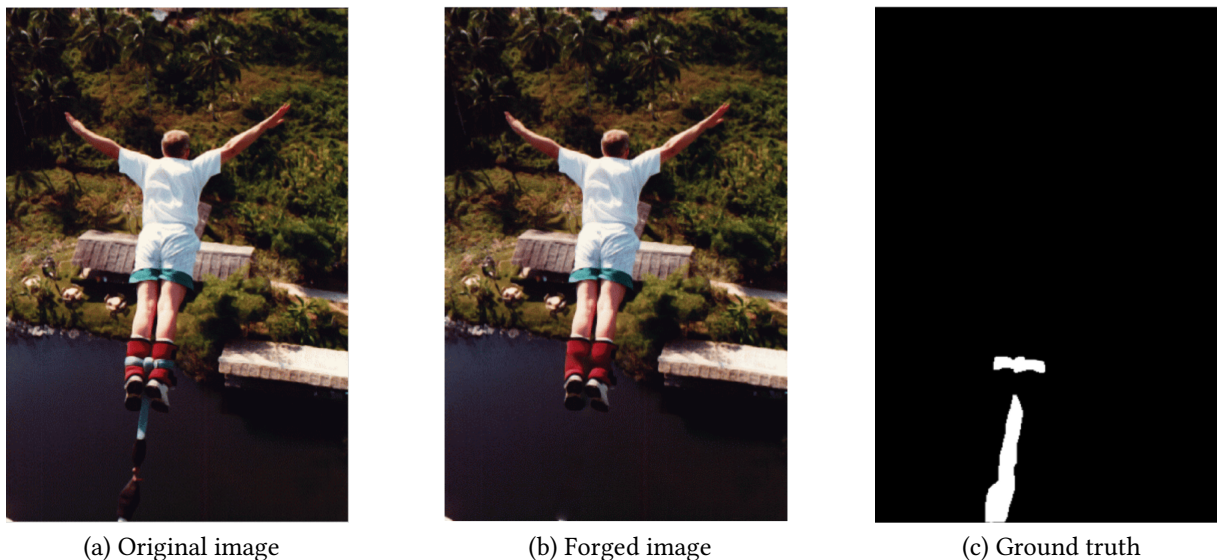


Figure 2.14: Inpainting for covering attack taken from [17]

exemplar-based. The methods of the first category exploit smoothness priors and partial differential equation to propagate local image texture from the exterior to the interior of the hole. A typical example of diffusion-based methods is the algorithm proposed by Bertalmio *et al.* in [17]. Meanwhile, the exemplar-based methods rely on image statistics and similarity priors to synthesize the patches in the unknown region. The works [46, 215] are some examples of this second category. Nowadays, these two methods are usually combined with deep learning (see e.g., [95, 184] for some recent reviews) to further improve the visual quality. Although sharing the same tampering attack as the cloning and splicing, the inpainting is obviously more sophisticated, and hence much harder to be detected.

2.3.4 Resampling

In image forgery, resampling is a technique that utilizes interpolation algorithm to maliciously geometrically transform a digital image or a portion of an image. It involves affine transformations like

resizing (see Figure 2.15) and rotation (see Figure 2.16). Such a forgery technique can be summarized



Figure 2.15: Resampling with resizing attack taken from the Image Manipulation Dataset [41]



Figure 2.16: Resampling with rotation attack taken from the Image Manipulation Dataset [41]

by two basis operations [169]: (i) *spatial transformation of coordinates* to move some parts or the whole original image onto a new sampling lattice, and (ii) *interpolation* to introduce specific correlations between neighboring pixels. Very often, the resampling happens together with other forgery techniques such as cloning or splicing to create a convincing forged images. This explains why many researchers have considered the resampling as an useful auxiliary trace to detect other forgeries (see e.g., [35, 156])

2.4 Passive Detection of Digital Image Forgery

Passive forgery detection is the family of forensic techniques that only utilize the received image to verify its authenticity or integrity [20]. Since no prior information possibly embedded in the original image, such as *authentication watermarks* or *digital signatures*, is available in hand, the passive techniques attempt to exploit inconsistencies existing in the digital image processing pipeline to detect forgeries. Such an approach is proved feasible and efficient [136, 171], because any manipulation likely disturbs the underlying statistics property or image consistency of a natural scene, and leaves traces in different stages of the processing pipeline [114]. Our review in this section focuses on the detection methods using in-camera and post-camera traces represented in Figure 2.17. The aim is to answer two questions: (i) *where these traces come from?*, and (ii) *what are the relevant methods for their detection?* Note that we will not cover all the state-of-the-art, but rather provide an essential summary

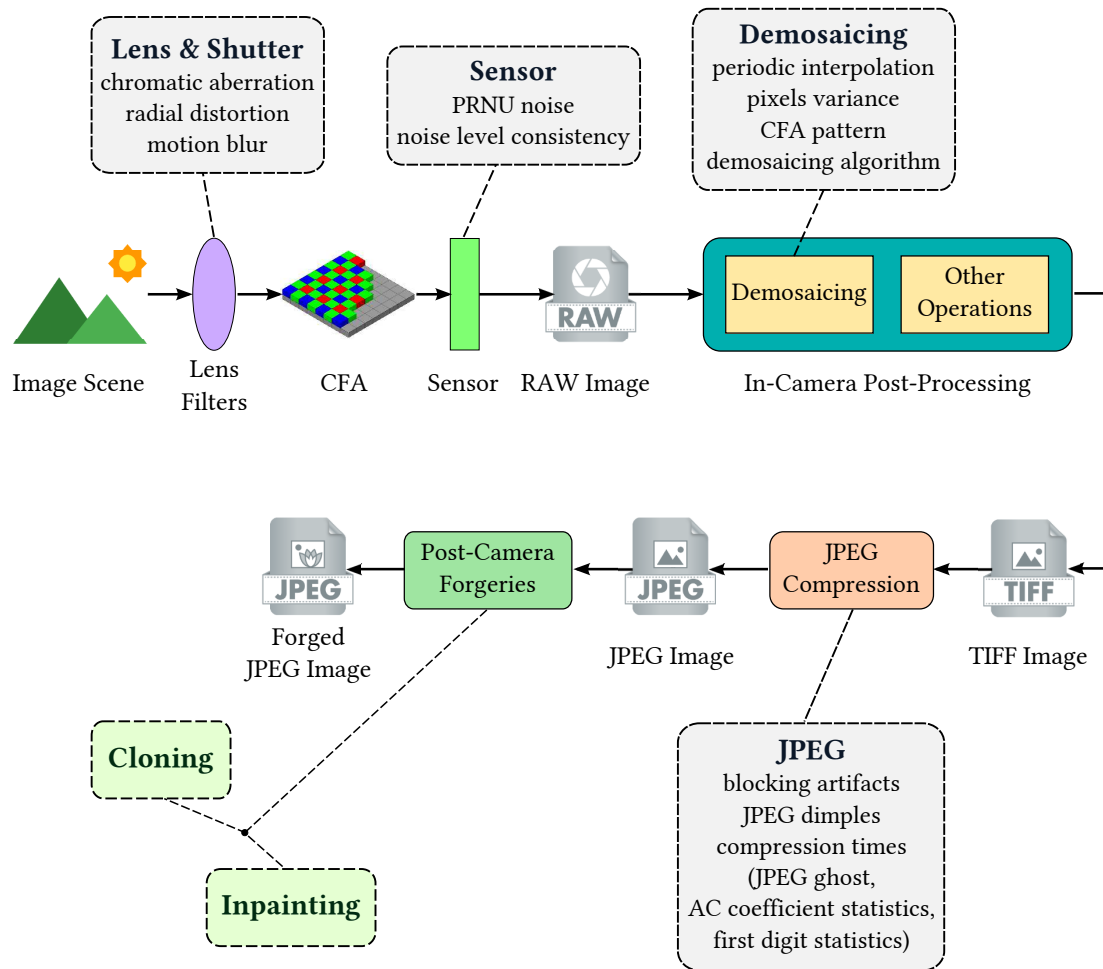


Figure 2.17: Overview of operation traces along with a typical digital image processing pipeline

of the research field. Existing works related to more specific problems studied in this thesis will be further introduced and analyzed at the beginning of Chapters 3, 4 and 5.

2.4.1 In-Camera-Based Detection

We have seen in Section 2.2 that each component and processing step in a digital camera acquisition process alters the input and leaves intrinsic traces in the image output. Moreover, since the optical module, image sensor and software of each camera device are customized by manufacturers, these traces vary across different camera brands and/or models. Their use as evidence for tampering detection is thus possible, independently of forgery types (see Section 2.3).

2.4.1.1 Lens and Shutter Traces

The integrity of image content can be exposed by inconsistencies in the lens and shutter parameters of the employed camera device. They include *lateral chromatic aberration* [150], *radial lens distortion* [219], and *motion blur* [187]. Since all of these techniques look for local deviations from a general

expected model, they are capable of localizing the forgery automatically.

Lateral chromatic aberration (LCA) arises due to the lens inability to focus all wavelengths of a single light ray to a single location on a sensor (see Figure 2.18a). Consequently, the focal locations



Figure 2.18: Illustration of chromatic aberration, fringing artifacts, and patterns inconsistency

of different wavelengths are displaced from each other in the image, and hence lead to the so-called *purple fringing artifacts* [105] as illustrated in Figure 2.18b. Johnson and Farid are perhaps the pioneers who used LCA to detect image forgeries in [98]. 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. The same algorithm is next applied to non-overlapped blocks to estimate the local patterns. The inconsistency between the global and local estimates are the evidence of forgeries (see Figure 2.18c for an example). Later, Yerushalmy and Hel-Or have shown in [225] that purple fringing artifacts can also be exploited for the image forgery detection. Although having a much more complex origin, this artifact is stronger and more visible than LCA. Again, inconsistency in the direction of these artifacts is used for tampering detection. More recently, Mayer and Stamm have proposed in [150] a statistical model to capture the inconsistency between global and local estimates of LCA. Based on this statistical model and hypothesis testing, the authors successfully derived an accurate statistic for forgery detection.

As depicted in Figure 2.19a, the differences in magnification level across a lens surface and the misalignment between lens and the detector plane cause a subtle radial shift in magnification towards the center of the lens. It results in the so-called *radial lens distortion* that makes straight edges in a scene

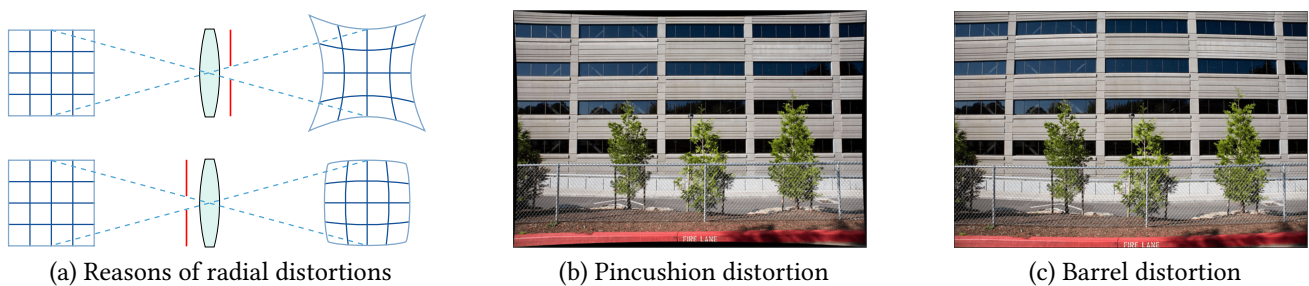


Figure 2.19: Illustration of radial lens distortions

appear curved edges in an image (see Figures 2.19c and 2.19b). Such an artifact is almost inevitable

and specific for different camera lenses [87]. This is why Choi *et al.* proposed to analyze this artifact as a fingerprint to identify the source camera in [40]. Using the Devernay's line extraction method [54], the authors estimate the distortion parameters of a camera, and then measure the error between the distorted line segment and the corresponding straight lines. The estimated parameters are next used to train a classifier to distinguish among images captured by different cameras. Chennamma and Rangarajan further applied the radial distortion to detect splicing forgery in [36]. The inconsistency in the radial distortion parameters across an image is the cue for the detection. Lately, in [71], Fu and Cao combined the radial distortion artifacts with a fake saliency map to improve the forgery localization. Figure 2.20 illustrates some results.

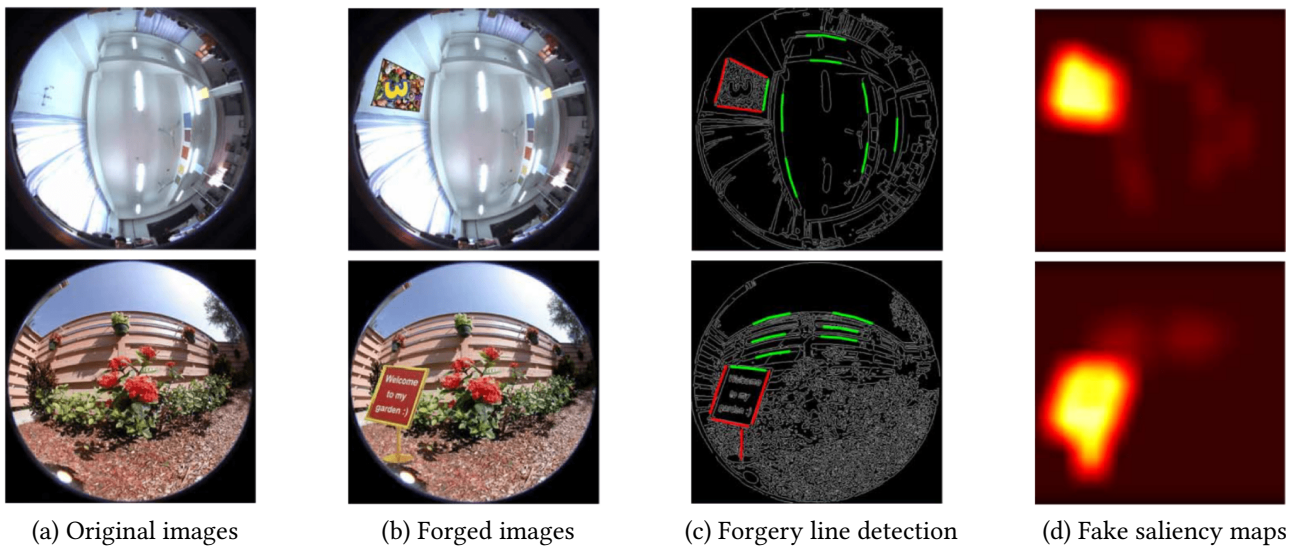


Figure 2.20: Forgery detection results extracted from [71]

If the two above traces are generated by the lens imperfection, the *motion blur* is mainly caused by the slow speed of the camera shutter relative to the object being imaged [171]. Especially, it is a very common phenomenon in images taken by hand-held cameras, and can thus serve as a fingerprint of natural images. In [101], Kakar *et al.* have relied on gradient analysis to estimate the motion blur direction (see Figure 2.21a). Meanwhile, Bahrami *et al.* have focused on the estimation of blur kernels in [10] (see Figure 2.21b). The differences in the blur direction or in the blur kernel between the whole

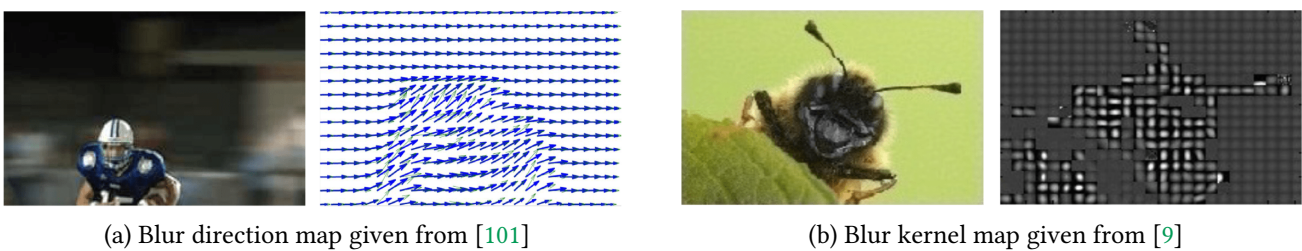


Figure 2.21: Illustration of local motion blur estimates

image and non-overlapping blocks therein allows to identify forged areas of the image.

Generally, we can find that the lens and shutter traces are not significantly applicable to the literature of forgery detection, because they are easily removed by a photo editing software or directly by the camera itself. For instance, the free software [RawTherapee](#) (GPLv3) allows to efficiently erase all the traces of lateral chromatic aberration, radial lens distortion and motion blur in an image.

2.4.1.2 Sensor Traces

As a dominant component of sensor pattern noise, the *PRNU noise* mainly arises from the heterogeneity of silicon wafers introduced during the sensor manufacturing process, and manifests itself as the pixel-to-pixel variation in light sensitivity. It is present in every natural image and practically unique to each sensor. These properties make the PRNU noise well adapt for the forgery localization. Its fundamental idea is to analyze the local correlation of PRNU signatures to identify specific areas with potential mismatch [34]. Figure 2.22 shows an example of tampering localization results for different variants of local signature analysis: (i) standard correlation field, (ii) segmentation-guided correlation field, (iii) adaptive-window correlation field, and (iv) peak-to-correlation energy field. We can remark

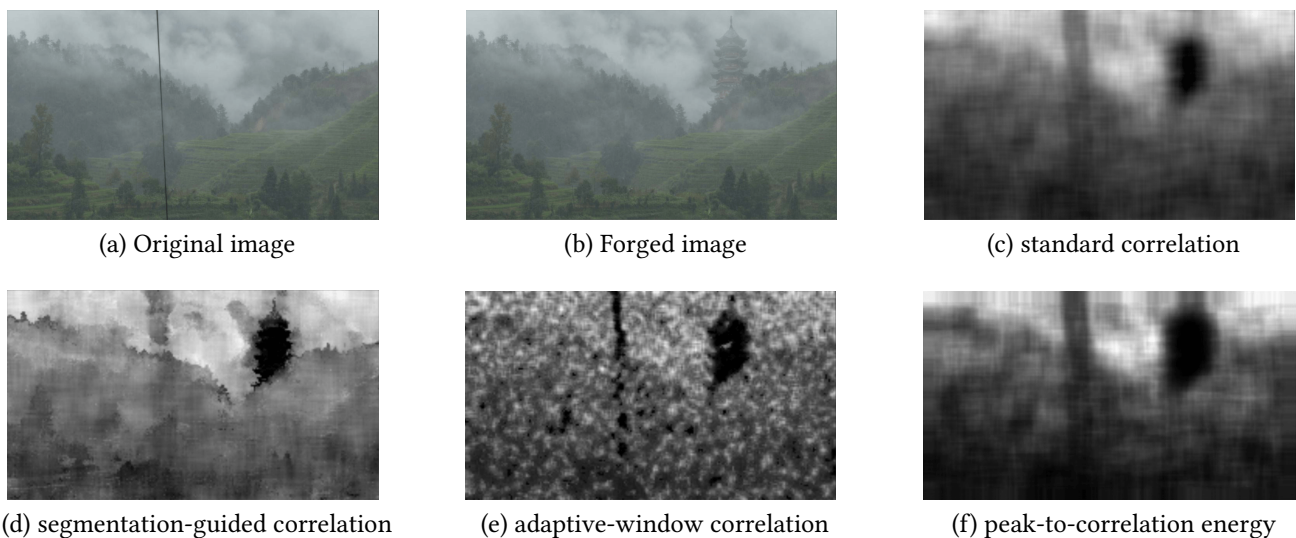


Figure 2.22: Example of tampering localization results using PRNU noise adapted from [115] and [114]

that the localization works poorly for small forgeries in highly-textured dark areas because of the very weak energy of PRNU noise. A way to remedy this flaw is to adopt better techniques for sensor noise estimation. A recent empirical evaluation of various techniques is reported in [4]. Another way is to improve localization capabilities for small forgeries. Indeed, recasting the PRNU-based localization problem in a Bayesian framework, modeling the decision variables as a Markov random field, and thus accounting for their spatial dependencies, Chierchia *et al.* [37] provide more accurate localization results than the approach developed by Chen *et al.* in [34]. More recently, Korus and Huang have conducted multi-scale analysis in [115] to further enhance the localization performance.

In addition to PRNU noise, *noise level*, which is a joint effect of the sensor characteristics, the current ISO setting, and prospective post-processing, is a potential cue for splicing detection, because it is usually different from an image to another (see e.g., Figure 2.23a). In [148], Mahdian and Saic

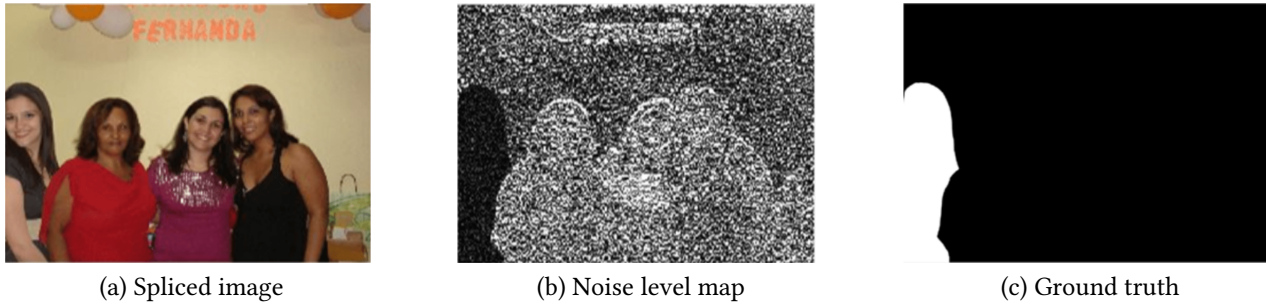


Figure 2.23: Example of spliced image, noise level map and ground truth extracted from [94]

estimated the local noise variance of image blocks based on the median absolute deviation in the high-pass Haar wavelet subband, and then used their inconsistencies to locate the splicing regions. A different method was introduced by Lyu *et al.* in [144], where block-wise noise estimation is based on the observation that the kurtosis values across different band-passed filter channels are constant. The method concludes by segmenting the image into regions with significantly different noise variances using k-means clustering. If the above works assumed the additive white Gaussian noise model which is independent of image content, the recent work of Yao *et al.* [223] have taken into account intensity-dependent noise. The authors first propose a noise level function that fit noise characteristics with possible variations of standard deviation with image intensity. Next, the noise level function is applied for both edge and non-edge regions of an image. By comparing the function values with a predefined threshold, tampered and original regions of a forged image can be divided. No matter what intensity-independent or dependent noise used, the splicing detection-based on the noise level is very sensitive to variations in the local noise variance. So, if the splicing is carefully prepared to keep the similar noise level in the whole forged image, forgeries will not be detected.

2.4.1.3 Demosaicing Traces

As mentioned in Subsections 2.2.1 and 2.2.2, most camera devices use a CFA together with a CCD or CMOS sensor chip to record image scene mosaically, and then interpolate missing colors by a demosaicing algorithm to achieve a full color image. The periodic pattern of CFA and unified demosaicing algorithms applied throughout color channels lead to *periodic interpolation artifacts* inherent in a natural image. Any local perturbation of this periodicity will put image integrity in doubt. Assuming a linear interpolation kernel, Popescu and Farid used in [174] an expectation-maximization algorithm to build a so-called *p-map* representing the correlation probability between a pixel and their neighbors. Such a *p-map* exhibits a periodic behavior which is clearly visible in Fourier domain. By applying this analysis to different areas of an image, forgeries can be detected. Figure 2.24 illustrates a detection result returned by this method. Gallagher and Chen proposed in [74] another way for detecting the periodic interpolation artifacts. Observing that the variance of the second derivative of an interpolated signal is periodic [73], the authors successively applied the high-pass filtering and the Fourier analysis to capture the presence of periodicity in the variance of interpolated/acquired coefficients. Compared to the method developed by Popescu and Farid, this method does not require the estimation

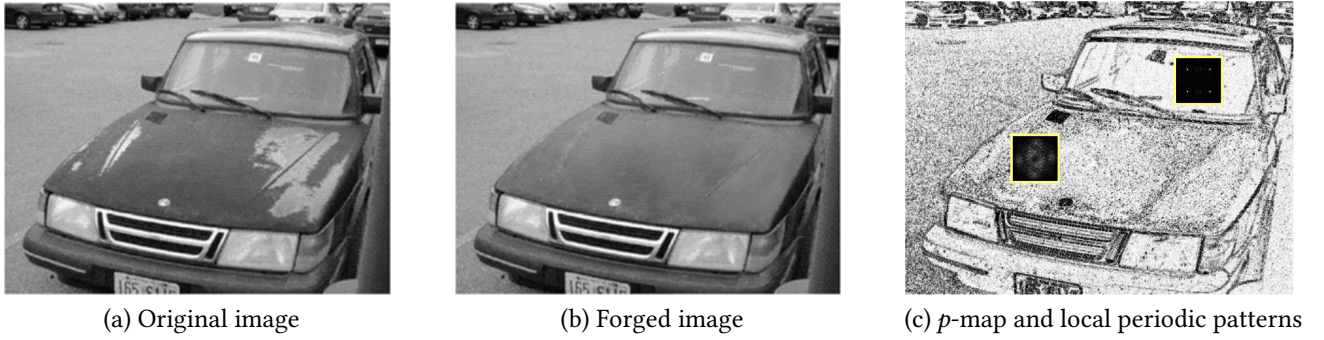


Figure 2.24: Illustration of p -map and local periodic patterns in Fourier domain adapted from [174]

of the CFA interpolation kernel, and hence is with lower computational complexity.

Along with the periodic interpolation, the *difference in the distribution of acquired and interpolated pixels* is also a fingerprint of natural images. In fact, since the missing pixels are interpolated from their acquired neighbors following an interpolation algorithm, the variance of interpolated pixels would be smaller than the acquired ones. Especially, Gallagher et Chen proved in [74] that this variance ratio equals $\frac{1}{4}$, when the bilinear interpolation is applied for the green channel (see Figure 2.25). Based

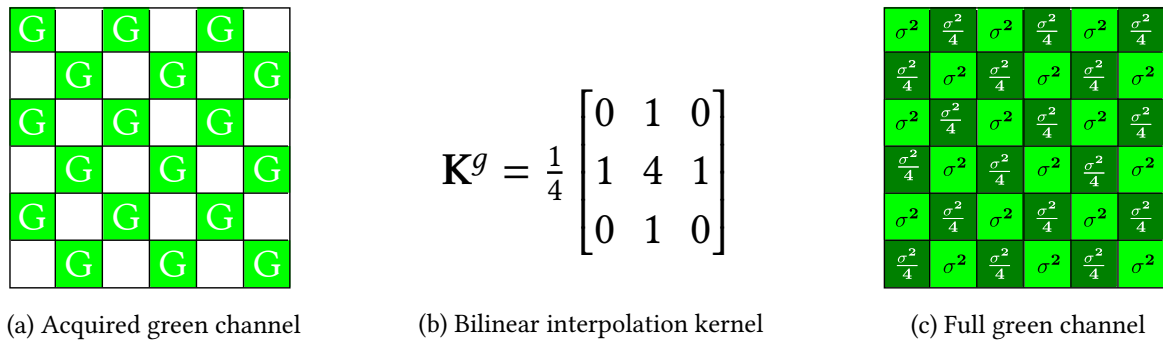


Figure 2.25: Variance of pixels in the green channel interpolated by bilinear kernel

on this phenomenon, Dirik and Memon [55] partitioned the considered image into non-overlapping blocks, and computed the associated ratio between variances of prediction residues in interpolated and acquired pixels. Since malicious operations likely erase this variance difference, the blocks whose variance ratio is close to 1 are the tampered ones. A similar method was proposed by Ferrara *et al.* in [63], but a Gaussian mixture model was employed to fit the distribution of prediction residues features rather than merely considering their variances. A naive Bayesian rule was next used to compute the tampering probability.

Besides, we can also use the *CFA pattern* and the *demosaicing algorithm* as proofs for the forgery detection, because they are specific for each class/branch/model of camera devices. In [55], Dirik and Memon have relied on an estimation of CFA pattern number to detect the presence of global or local tampering in an image. Swaminathan *et al.* exploited in [200] the inconsistencies among the estimated demosaicing coefficients to check if an image has undergone any form of subsequent processing.

Despite very precise localization ability, demosaicing artifacts constitute a relatively simple fea-

ture which lacks both robustness and security. For instance, traces of periodic interpolation are easily destroyed by JPEG compression, even with the highest quality levels. Hence, this trace is applicable almost exclusively for uncompressed images. Moreover, since existing methods assume that the camera uses the standard Bayer CFA and a non-adaptive demosaicing algorithm, false positive errors or unreliable localization maps may appear when these conditions are violated.

2.4.1.4 JPEG Compression Traces

JPEG is no doubt the most popular format of digital images today. This makes JPEG compression traces one of the most important tools for image forgery detection. In the following analysis, we will distinguish traces in *spatial domain* from traces in *DCT domain*.

Generated by the block-wise processing, *blocking artifact* is a well-known fingerprint in spatial domain of JPEG compressed images. It corresponds to the discontinuities that regularly occur along the boundaries of every 8×8 pixel block (see Figure 2.26 for an illustration). In the presence of forgery



Figure 2.26: Illustration of JPEG images and associated maps of block artifacts given from [160]

operations such as cloning or splicing, this regular structure is locally disturbed by the block grid of inserted objects, and some misalignments likely appear in the forged image. Fan and de Queiroz proposed in [57] a lightweight algorithm to check whether an image is JPEG compressed and to further locate the whole position of block artifacts. The algorithm is based on the idea that if the image has not been compressed, the pixel differences across 8×8 block boundaries should be similar to those within blocks. Therefore, it is possible to evaluate the difference between of inter-block and intra-block pixels, and then define a threshold to detect the presence of prior JPEG compression. The algorithm performance is recently improved in [93] thank to the content-aware detection of grid inconsistencies. In [135], Li *et al.* developed another algorithm to blindly extract the blocking artifact grid in JPEG images. The algorithm automatically generates a tampering map indicating where the local blocking grid diverges from the global one.

Beside block artifact, *JPEG dimple* is another trace of JPEG compression in spatial domain. It is issued from the different rounding operations (e.g., round, floor, or ceiling) possibly used to convert DCT coefficients from floating-point to integer values [2]. In fact, the use of floor and ceiling can lead to a periodic artifact in the form of a single darker and brighter pixel in the top-left corner of 8×8 pixel blocks (see Figure 2.27). The local absence of such dimples in a JPEG image would be the evidence of forgeries. Agarwal and Farid have proved in [3] that JPEG dimples allow to detect a wide range of manipulations such as cloning, splicing, median filtering, resampling, as well as content-aware fill.

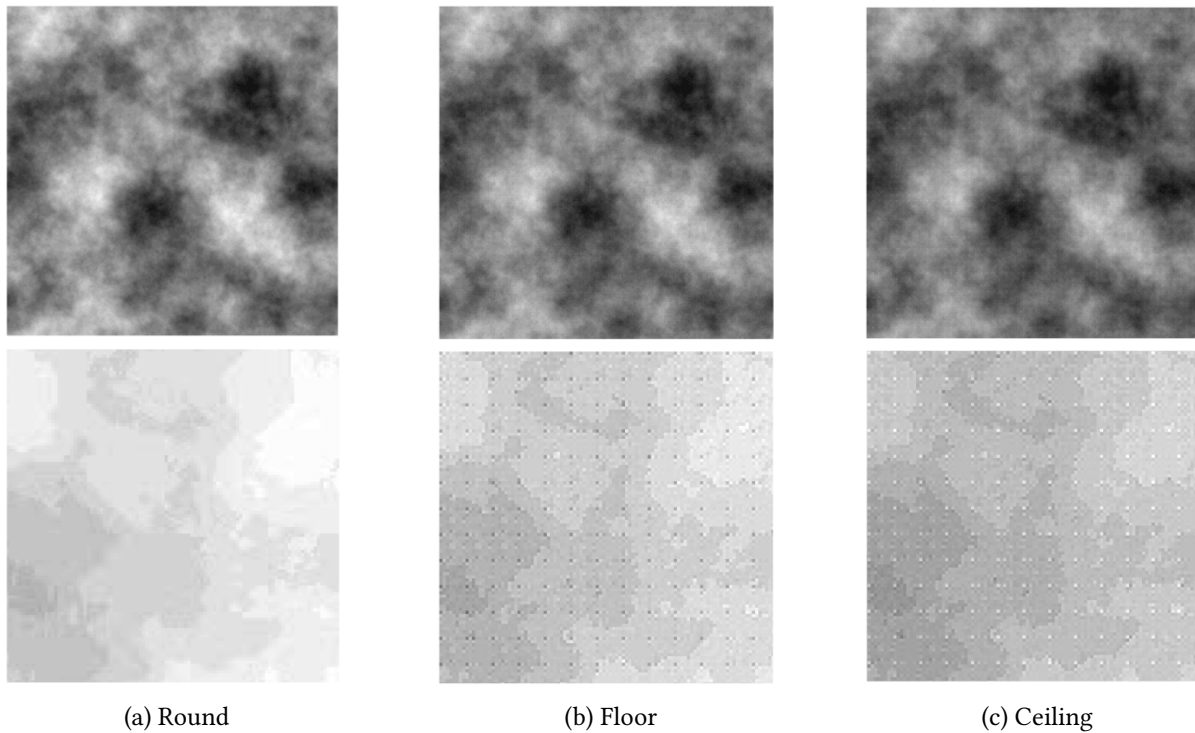


Figure 2.27: Illustration of JPEG duple artifacts taken from [2]

Very often, the tampering process of digital image finishes by resaving forged images under JPEG format. This implies that the *multiple JPEG compression* is a potential fingerprint of forged images, and can thus serve forgery detection. In spatial domain, tampered areas in an image manifests as the *JPEG ghosts* appeared after recompressing the forged JPEG image following various lower quality factors. As displayed in Figure 2.28, the nearer the quality factor of recompression to the original factor of tampered areas, the more the JPEG ghost is clearer. Such a forgery detection was first introduced

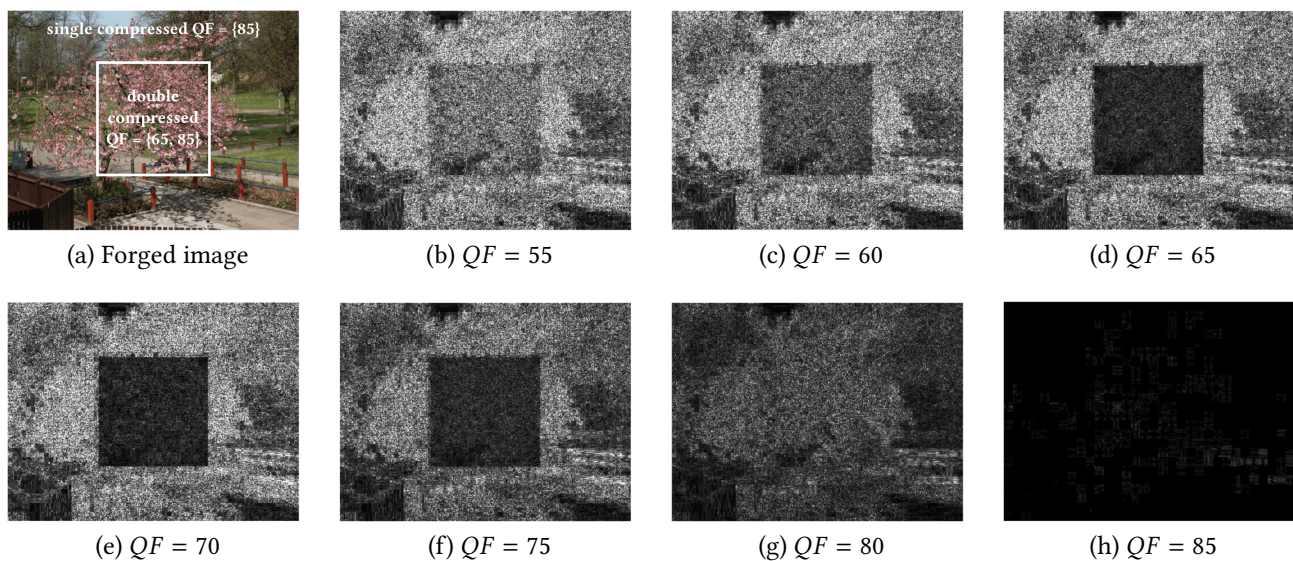


Figure 2.28: Difference between tampered image and its re-saved versions adapted from [62]

by Farid in [62], and afterward improved by Zach in [227] and by Zhang and Wang in [231]. In DCT domain, the multiple JPEG compression usually leaves traces in the *histogram of some selected DCT coefficients* [18] or of *their first digit* [155]. We can remark from Figure 2.29a that peaks or valleys in

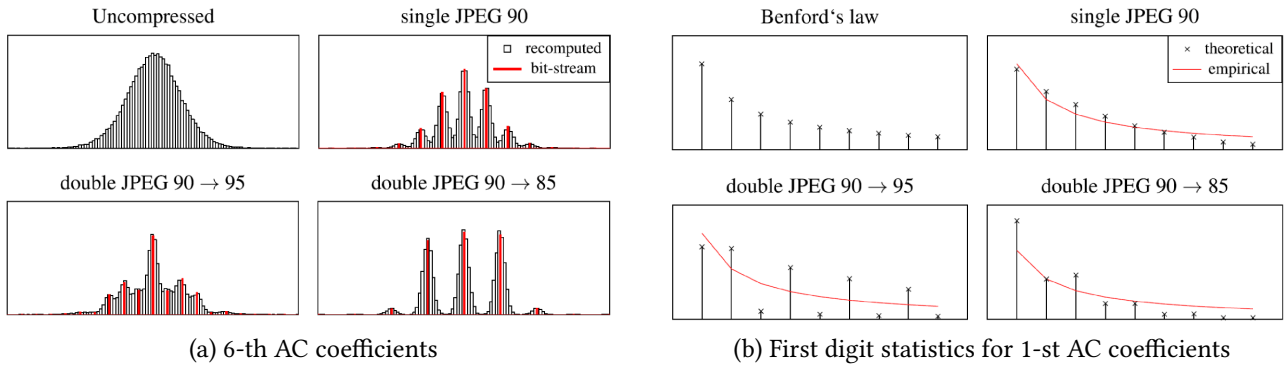


Figure 2.29: Histograms of DCT coefficients statistics in the luminance channel adapted from [114]

the histogram of AC DCT coefficients characterize JPEG compression times differently. This conducts various features for forgery detection. In [138], Lin *et al.* exploited the periodic peaks and valley in the histogram of DCT coefficients for double compression detection, and then used this feature to automatically locate tampered regions. This work was subsequently improved by Bianchi *et al.* who recognized that the empirical distribution is essentially a mixture of an authentic and altered components [19]. The approach was further extended in [18] by separately considering cases of aligned and misaligned JPEG blocking grid. Another efficient tampering localization approach is to based on the first digit distribution of DCT coefficients. As depicted in Figure 2.29b, this distribution is well fitted by the Benford's law [100] when the image is single JPEG compressed, and becomes much worse when the image is double compressed. This observation leads to *global first digit features* in [71], and *mode-based first digit features* in [130]. These features was next combined with a multi-class classification to detect double compressed JPEG images. Lately, Milani *et al.* extended this approach to the detection of higher compression on JPEG images in [155].

Generally, existing JPEG-based methods for tampering localization work well if the second quality level is greater than for the original image. If the quality level difference is sufficient, reliable localization can be performed even for 8×8 pixel blocks. However, this performance drops significantly when the second quality factor is equal or lower than the original one, and much larger windows are required for adequate results [116].

2.4.2 Post-Camera-Based Detection

Various post-camera manipulation applied to the digital image, even if not visually detectable, modify the image properties and leave peculiar traces accordingly to the processing itself. This section discusses how such specific traces can be exploited to detect the *cloning* and the *resampling*.

2.4.2.1 Cloning Traces

In accordance with the feasibility and popularity of the malicious cloning, its detection is one of the most active research topics (see e.g., [203] for a thorough review). Here, we briefly discuss two generic classes of cloning detection referred as *patch-based algorithms*⁶ and *keypoint-based algorithms*.

From the mechanism of the cloning attack, the idea of *patch-based* detection algorithms is to consider a suspicious image as small patches, and try to match each patch to the others based on some compact content representation. A cluster of neighboring patches matched to the same source region indicates a potential cloning. A common workflow of such algorithms is depicted in Figure 2.30. Fridrich *et al.* are the pioneers in applying this workflow for cloning detection in [69]. They suggested looking for matches among DCT coefficients of overlapping image blocks. To reduce the computa-

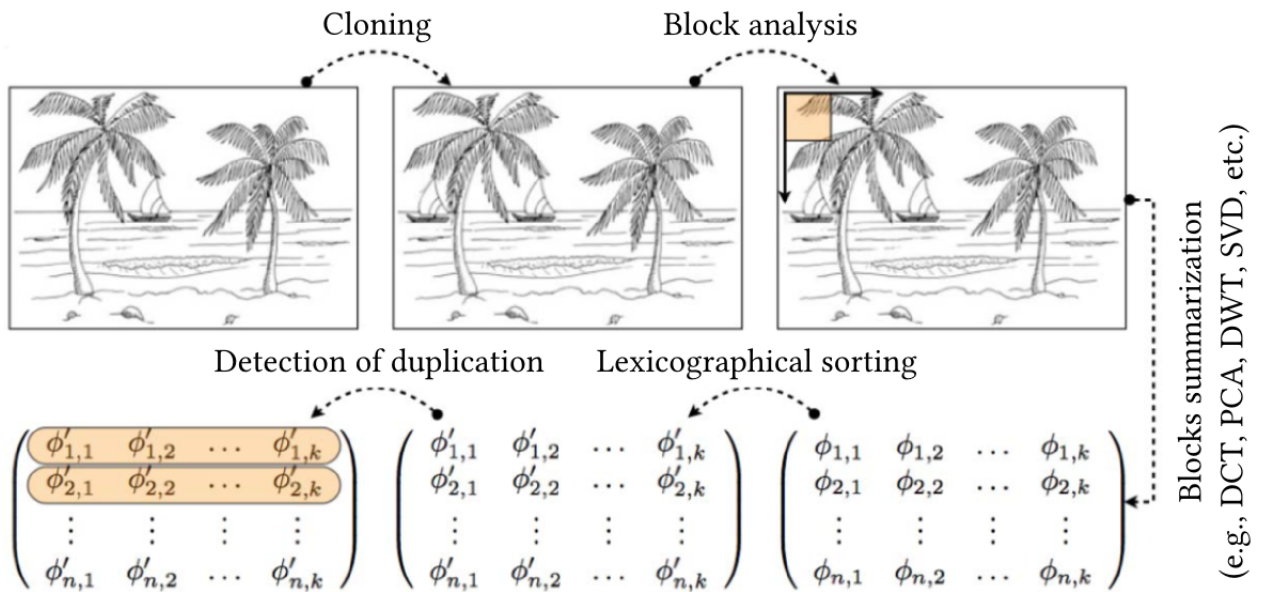


Figure 2.30: Workflow of patch-based algorithms for cloning detection extracted from [64]

tion cost and the comparison complexity, the matrix of DCT coefficients is lexicographically sorted, and two adjacent identical lines are considered as cloned regions. Since then, various improvements have been proposed, where the main difference resides in features used to match the blocks. Indeed, instead of DCT matching, we can also use principal component analysis [173], discrete wavelet transform [229], singular value decomposition [102], Fourier-Mellin transform [16], Zernike moments [191], etc. Albeit these efforts, existing patch-based algorithms are always faced with considerable computational complexity.

To remedy the computational drawback of patch-based algorithms, *keypoint-based algorithms* attempt to use some characteristic points in patches to estimate their similarities rather than the whole image path (see Figure 2.31 for an example). In the literature, such keypoints are essentially built from the scale-invariant feature transform (SIFT) [140] and the speeded-up robust features (SURF) [13]. Given these features, the variety of keypoint-based algorithms comes from different choices of matching policies. For instance, the authors of [91, 163] used similarity search of SIFT keypoints

⁶Also known as *block-based algorithms*.

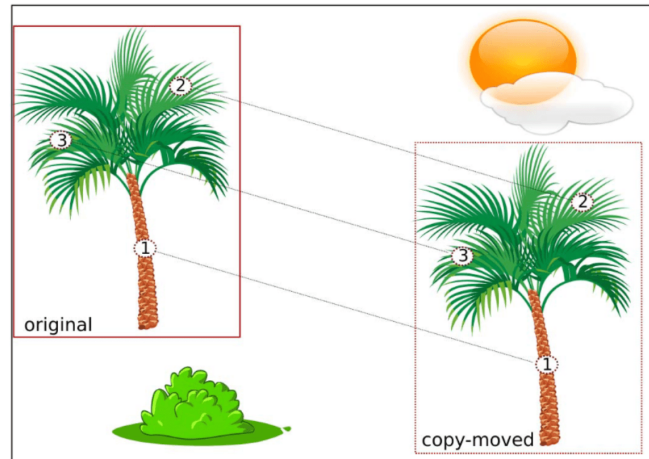


Figure 2.31: Illustration of keypoint-based algorithms for cloning detection extracted from [64]

descriptors. Amerini *et al.* analyzed SIFT correspondences by means of hierarchical clustering procedure in [6]. Shivakumar and Baboo combined SURF with a KD tree to enable a cloning detection of high resolution images with a minimum number of false matches. Regardless of chosen algorithms, keypoint-based detectors work poorly for small and homogeneous cloned areas, because keypoints therein usually are not enough to give a robust detection.

In principle, detectors based on cloning traces is incapable of distinguishing between either naturally occurring self-similarity or even the original and the cloned objects.

2.4.2.2 Resampling Traces

As mentioned in Subsection 2.3.4, the image resampling always introduces specific *correlations* between neighboring pixels. Moreover, if the image is resampled with respect to a single ratio factor, these correlations will be *periodic* in nature. Based on such traces, various demosaicing detection algorithms can be reused to detect the resampling.

Using the same expectation-maximization algorithm as in [174], Popescu and Farid successfully estimate the linear interpolation kernel employed for the resampling, as well as the p -map representing the correlations among pixels. Periodicities in the p -map can be identified by observing distinct isolated peaks inherent in its Fourier transform (see Figure 2.32 for some examples).

Almost in parallel with the work of Popescu and Farid, Gallagher observed in [73] that the variance of the second derivative of an interpolated signal is periodic. The periodicity in the second derivative of the overall image is found by analyzing its Fourier transform. Although derived from different bases, Kirchner *et al.* proved in [106, 110] that the Popescu's method and Gallagher's method are closely related. In these papers, it is demonstrated how the variance of prediction residuals of a re-sampled signal can be used to described periodic artifacts in the corresponding p -map. It is also proposed a simplified detector, much faster than the one in [175], while achieving similar performance. Further studies by the same authors are reported in [107, 108]. Based on Gallagher's ideas, the periodicity of the second derivative (or other order) is further analyzed by Prasad and Ramakrishnan in [176], by Mahdian and Saic in [146], by Dalgaard *et al.* in [49] and by Song *et al.* in [197].

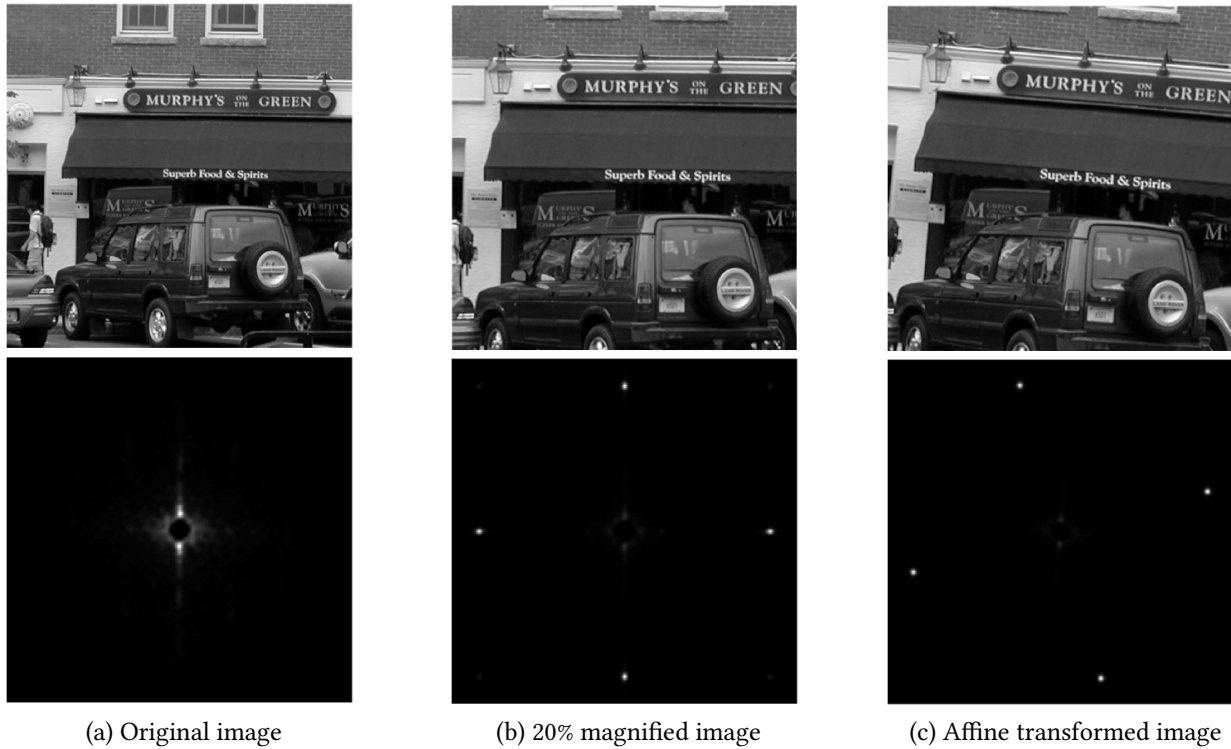


Figure 2.32: Images and the Fourier transform of their p -map adapted from [175]

Another approach to resampling detection has been developed by Mahdian and Saic [147] on the basis of the periodic properties of the covariance structure of interpolated signals and their derivatives. The core of the proposed scheme is a Radon transform applied to the derivative of the investigated signal, followed by a search for periodicity. Always exploiting the covariance structure of interpolated signals, Pasquini and Böhme quantified in [176] the statistical distance between an original signal and its downsampled version by means of the Kullback-Leibler divergence in case of a wide-sense stationary first-order autoregressive signal model. In [25], the authors employed an artificial neural network with two hidden layers to perform a binary classifier characterizing resampling. More recently, Qiao *et al.* investigated in [181] the problem of image re-sampling detection based on the linear parametric model. The authors proposed to estimate the probability of pixels noise, and thence to design a practical likelihood ratio test (LRT) to detect re-sampled images. They argued that the designed LRT-based detector is the first test based on hypothesis testing theory for image resampling detection, which can achieve the maximal detection power at the prescribed false alarm rate. Besides, the detector can deal with the problem of authenticating the resampled images from mixed compressed and uncompressed images.

2.5 Conclusion

We have reviewed and analyzed in this chapter the image formation process inherent in a typical digital camera, four well-known techniques for image forgery, and various passive methods for image

forgery detection based on traces left by in-camera and post-camera processing steps. Such a review and analysis allows to specify better the problems to be dealt with in this doctoral project. Indeed, aiming at versatile tools, we are interested more especially in exploiting traces returned by the demosaicing, the JPEG compression and the resampling post-processing operation to develop forgery detectors. This choice can be explain by upstream operations of the demosaicing and the JPEG compression and by the necessity of the resampling after any other forgery technique, so that the developed methods can detect various forgery types. The detail development of these forgery detectors are further represented in Chapter [3](#), [4](#) and [5](#) respectively.

Chapter 3

Digital Image Authentication and Forgery Localization Using Demosaicing Artifacts

3.1 Introduction

In a digital camera acquisition pipeline, demosaicing (also known as color filter array (CFA) interpolation) serves to reconstruct a full color image from the sampled data overlaid with a CFA (see Figure 2.1). Although most of commercial camera devices share a common acquisition pipeline process, each step therein is personalized according to manufacturer choices. Traces left by demosaicing are thus different but specific for different camera brands and/or models. When these traces are totally missing in an image, or when there exists any inconsistency in traces for different image regions, the photographic image is likely to be tampered. Moreover, since the demosaicing is an upstream operation in the acquisition pipeline of digital cameras, its traces are resistant to post-camera manipulations such as cloning, splicing, inpainting, resizing, etc. Therefore, the demosaicing traces could be a robust and powerful evidence to assess the credibility of digital images. This versatility motivates us to develop in this chapter a demosaicing traces-based algorithm for image authentication and forgery localization.

3.1.1 Prior Art

Although an overview has been briefly done in Section 2.4.1.3 of Chapter 2, its specific focus on the demosaicing traces used for forgery detection seems not enough to highlight the originality of the algorithm proposed in this chapter. A more detail literature analysis with new categorization is thus proposed here. Indeed, we can further categorize forensics works considering demosaicing traces into two main classes. The first one consists of algorithms aiming at estimating from a digital image the CFA pattern and/or the demosaicing algorithm inherent in camera devices. The second one characterizes algorithms used to evaluate the presence/absence of artifacts generated by demosaicing operations.

3.1.1.1 First Class

As to the first stream in the class, the works [38, 96, 108, 193, 202] represent key methods for CFA pattern identification. In [108], the Bayer CFA pattern is identified by minimizing the difference between the raw sensor signal and the inverse demosaiced signal. In [38], the identification is performed via an intermediate value counting algorithm developed from the observation that the value of interpolated color samples is always between the minimum and maximum values of their neighbors. Other method is to compute the ratio between the average noise variance of interpolated pixels and of acquired pixels for all possible candidate CFA patterns of a digital image; the pattern providing the largest ratio is considered as the true one [202]. More recently, color difference blocks are proposed as a means for estimating the CFA configuration in [96, 193].

Beyond the CFA pattern identification, works within the second stream focus more especially on estimating demosaicing algorithms. In [174], an expectation-maximization (EM) algorithm is employed to estimate the coefficients of linear interpolation kernels. An improvement is made in [15] by combining the EM algorithm with average second-order derivative spectrum [73] to obtain interpolation coefficients from smooth and non-smooth regions of images separately. Also partition an image into smooth, horizontal and vertical non-smooth regions, a two-step estimation process is proposed in [199]: (i) linear interpolation coefficients associated with each of candidate CFA patterns are first derived using the singular value decomposition, (ii) a minimum interpolation error criterion is next used to jointly identify the correct CFA pattern and demosaicing algorithm for separate image regions. In [27], an accurate method based on the partial second-order image derivative correlation models is proposed to recover demosaicing formulas. The method allows to take into account the correlation between three color channels of the image.

Since the CFA pattern and/or the demosaicing algorithm are specific for each class/branch/model of camera devices, they can be used as proofs for image forgery detection and localization. For instance, in [28, 200], the inconsistencies among the estimated demosaicing coefficients are exploited to check if an image has undergone any form of subsequent processing. By remarking that the CFA pattern is changed if the image color is modified, the authors of [39] have designed an advanced intermediate value counting algorithm for measuring the change in the CFA pattern, and hence localizing the extent of color modification in digital images. We also note that estimated CFA pattern and/or demosaicing algorithm are used not only for assessing the credibility of digital images content, but also for camera source identification (see e.g., [75]). However, the latter is out of our scope.

3.1.1.2 Second Class

The first stream in the second class relies on periodic artifacts caused by the demosaicing. In fact, the image sensors in a CFA are usually organized periodically [153]. Besides, many demosaicing algorithms behave as a filtering process where missing signals are interpolated by periodically applying an interpolation kernel to acquired signals (see e.g., [174]). Therefore, periodicity in demosaicing artifacts is intrinsic to digital photographs, and may help to authenticate images. Inspired by this idea, Popescu and Farid build in [174] a probability map to expose the periodic pattern of correlated pix-

els. When interpolated pixels are present, the periodicity of the map is clearly visible in the Fourier domain. Such an analysis can be applied to different areas of the tested image to detect the presence of local tampering, however the area size should not be smaller than 256×256 to assure the accuracy of the results. Observing that the variance of the second derivative of interpolated images is periodic [73], Gallagher and Chen use in [74] high pass filtering and Fourier analysis to compute periodic frequency locally. Forged regions of an image will have a different periodicity than the rest. This method is applicable for image blocks with size 64×64 or more. Another method based on periodicity analysis of prediction errors variance is recently proposed by Li *et al.* in [134]. Assuming that interpolation errors are Gaussian distributed, a posterior probability map is derived according to Bayes' rule. Such a map shows the periodic interpolation of the tested image in an intuitive way. An analysis with two dimensional discrete Fourier transform allows to capture the periodicity. By experiments, Li *et al.* show that this method can be applied for local tampering detection with blocks size up to 32×32 .

For the second stream, methods for digital image forgery detection are based on the differences in the distribution of acquired and interpolated pixels. In [55], Dirik and Memon recognize that the low pass nature of CFA demosaicing make the variance of the sensor noise in interpolated pixels significantly lower than acquired pixels. As a result, demosaicing artifacts can be measured by a ratio of noise variances between interpolated and acquired pixels. If this ratio is close to 1, tampering has been performed on the image. The method presents a good performance for image blocks with size greater than 96×96 . Sharing the same idea, Ferrara *et al.* [63] carry out fine-grained analysis of CFA artifacts and propose a feature to measure the presence of demosaicing even at the smallest 2×2 block level. However, numerical experiments show that the tampering localization yields the best performance at 8×8 block size. While the above works consider spatial features of demosaicing artifacts, González-Fernández *et al.* [79] are rather interested in their spectrum. Indeed, by computing the probability of each interpolated pixel and then applying the discrete cosine transform (DCT) on small blocks of the probability map, the presence/absence of the demosaicing artifacts within a block could be verified via the DCT coefficient at the highest frequency. Experiments show that the method is reliable for blocks of size 16×16 .

Compared to the first stream, methods within the second stream brings out better localization resolution and higher fidelity. However, their performance seems more sensitive to JPEG compression than the first stream, especially when the compression quality is significantly low.

3.1.1.3 Common Remarks

Since the demosaicing is an upstream operation in the acquisition pipeline of digital cameras, its traces are independent of forgeries such as cloning, splicing, inpainting, resizing, etc. Therefore, methods based on demosaicing traces do not target any specific forgery operation, but are rather applicable to a variety of operations. Despite this advantage, demosaicing traces are easily destroyed by JPEG compression, even with very high quality levels. This is why these methods is suitable to uncompressed or less-compressed photographs. This is a common and almost unavoidable limitation of demosaicing traces-based methods. Moreover, very often a comparison threshold is required to detect or localize forgeries (see e.g., [55, 63, 134]). In practice, the choice of such a threshold is not easy and may be

very influential in the robustness of the methods. Besides, employed demosaicing traces (i.e., CFA pattern, demosaicing algorithm, periodicity of demosaicing artifacts, differences in the distribution of acquired and interpolated pixels) are closely correlated. However, the above classification of related works implies that they are treated separately in most existing algorithms. Jointly use these traces could improve the performance of image forgery detection and localization.

3.1.2 Contributions and Organization

The two classes of forensic algorithms using demosaicing traces have been separately studied in the literature. Here, we aim at combining both of them into an unified algorithm in order to improve the image authentication and forgery localization performances. As a result, a hybrid algorithm consisting of CFA pattern identification, demosaicing algorithm estimation, and artifacts analysis is developed. Aiming at fine-grained detection, we have adopted the identification scheme proposed in [199] to reveal the CFA pattern and interpolation kernel, and a local analysis similar to [63] to expose demosaicing artifacts from prediction residues. Despite these similarities, several improvements have been made in the algorithm.

1. We theoretically analyze how the mean and variance of prediction residues in interpolated and acquired signals evolve in a digital camera acquisition pipeline from the RAW format to the JPEG format. This is the basis to build feature statistics of interest. Besides, the analysis also helps to explain in part why the demosaicing traces-based approach is less effective with JPEG compressed images.
2. In most demosaicing traces-based algorithms (see e.g., [63, 134, 174, 194]), prediction residues are given from entire pixels. Here, by further partitioning a pixel into content and noise parts, we realize that the demosaicing behaves in the same manner for content, noise, as well as entire pixel. However, due to the weak energy of noise, demosaicing traces are most visible in noise part. This is why feature statistics used in the algorithm are extracted from noise residues rather than the residues of content or entire pixels.
3. As mentioned in [63], the presence of sharp edges in images reduces the quality of feature statistics, because it may disrupt the correlation between interpolated and acquired residues. To overcome this obstacle, we build our feature statistics from smooth region of the green channel instead of the entire channel. As such, the edges problem is no longer a great concern.
4. Regarding the authentication, after verifying the standard normal distribution of feature statistics in natural images, we adopt normality tests (i.e., Anderson-Darling test, one-sample Kolmogorov-Smirnov test, Jarque-Bera test, and Lilliefors test) to authenticate them automatically. Such an automatic detection is somewhat scattered in demosaicing traces-based works. Besides, we also provide other tools (i.e., Q-Q plot diagram, probability distribution curves, and localization map) which can help to improve authentication performance by human interpretation.

5. By modeling the distribution of feature statistics in tampered images as a normal mixture, we applied a *penalized Expectation-Maximization* (EM) algorithm to localize forged regions. Since the algorithm does not require a comparison threshold, it provides robuster localization results than traditional threshold-based methods [55, 63].

Numerous numerical experiments also confirm that the proposed algorithm yields better performance than existing algorithms of the same kind.

The remainder of the chapter is structured as follows. Section 3.2 focuses on analyzing statistical properties of demosaicing traces when they evolve in a digital camera acquisition pipeline. The analysis is next extended to develop feature statistics measuring the unbalance between the local variances of prediction residues for the green channel of demosaiced images in Section 3.3. A robust feature statistic is eventually identified. Section 3.4 develops a single statistical model for the proposed feature statistic. Such a model is next used to authenticate digital images and to localize the tampered regions. Numerous numerical experiments are provided in Section 3.5. Some conclusions are discussed in Section 3.6. Finally, an appendix on the formalization of penalized EM algorithm for model parameters estimation is provided in Appendix A.

3.2 Statistical Analysis of Demosaicing Traces

Analyzing statistical properties of demosaicing traces is a key step to define a relevant feature statistic for tampering detection and localization. In this section, we define a *signal* as either the *pixel*, its *content* part or its *noise* part, and we are interest more especially in the mean and variance of prediction residues in both interpolated and acquired signals of TIFF or JPEG images. In ideal cases, we prove that the residues variance in acquired signals is greater than in interpolated signals, while their mean values are always 0. However, such a difference weakens under the impact of JPEG compression. Hereinafter, for an easier representation, the analysis is just done for one dimensional signals (i.e., a row in the green channel of digital image). The results for two dimensional signals can be interpreted in the same way.

3.2.1 Demosaicing Traces at Pixel Level

Let consider a row with size N in the green color channel of an digital image obtained by using a *Bayer filter* and a *linear* demosaicing algorithm. Without loss of generality, we assume that the Bayer CFA pattern is arranged in the manner that acquired pixels correspond to even positions of the row (old positions corresponds to pixels to be interpolated from acquired pixels). Each acquired pixel consists of two parts: *real scene content* and *noise*. The real scene content is the true image information that we take and expect, while noise is generated everywhere during camera imaging. Mathematically, an acquired pixel $p_A(x)$ at position x on the row can be expressed as

$$p_A(x) = \begin{cases} c_A(x) + n_A(x) & \text{if } x \text{ even,} \\ 0 & \text{if } x \text{ odd,} \end{cases} \quad (3.1)$$

where $c_A(x)$ and $n_A(x)$ are respectively the content and noise of $p_A(x)$. Applying a linear demosaicing algorithm with interpolation kernel h_u to the acquired pixel, we obtain the resulting pixel

$$p_R(x) = \begin{cases} p_A(x) = c_A(x) + n_A(x) & \text{if } x \text{ even,} \\ p_I(x) = \sum_{u \neq 0} h_u p_A(x+u) & \text{if } x \text{ odd,} \end{cases} \quad (3.2)$$

where $\sum_{u \neq 0} h_u = 1$. The resulting pixel $p_R(x)$ is either an acquired pixel $p_A(x)$ or an interpolated pixel $p_I(x)$. Since only acquired pixels at even positions have values and contribute to the sum of interpolated pixels, $x+u$ in $p_A(x+u)$ is even, or equivalently u is odd. By substituting the first line of (3.1) in the second line of (3.2), we obtain

$$p_R(x) = \begin{cases} p_A(x) = c_A(x) + n_A(x) & \text{if } x \text{ even,} \\ p_I(x) = c_I(x) + n_I(x) & \text{if } x \text{ odd,} \end{cases} \quad (3.3)$$

where $c_I(x)$ and $n_I(x)$ denote respectively the content part and noise part of $p_I(x)$

$$c_I(x) = \sum_{u \neq 0} h_u c_A(x+u), \quad (3.4)$$

and

$$n_I(x) = \sum_{u \neq 0} h_u n_A(x+u). \quad (3.5)$$

The similar expressions of $p_I(x)$, $c_I(x)$ and $n_I(x)$ imply that the demosaicing alter the content part and noise part in the same way as the entire pixel. In other words, we can find traces left by the demosaicing in the content, in the noise and in the entire pixel at odd positions of the considered row.

Hereinafter, we use s to represent a signal which may be either the pixel, its content part or its noise part. Accordingly, s_A , s_I and s_R stand for the acquired, interpolated and resulting signals respectively, and they are linked by the following relation

$$s_R(x) = \begin{cases} s_A(x) & \text{if } x \text{ even,} \\ s_I(x) = \sum_{u \neq 0} h_u s_A(x+u) & \text{if } x \text{ odd,} \end{cases} \quad (3.6)$$

We will analyze the mean and variance of signal residue to find out some useful demosaicing artifacts when the digital image is in TIFF format and in JPEG format.

3.2.2 Demosaicing Artifacts in TIFF Images

As illustrated by Figure 2.1, the RAW image, after demosaicing, white balancing and gamma correction, becomes an uncompressed high-quality image in TIFF format. Since, the white balancing and the gamma correction are lossless operation in terms of information [207], we expect that the characteristics of demosaicing artifacts do not much changed. Let k_u , with $\sum_{u \neq 0} k_u = 1$, is an estimate of the interpolation kernel h_u when the considered image is in TIFF format, then the predicted signal is

computed as

$$s_P^{\text{tiff}}(x) = \sum_{u \neq 0} k_u s_R^{\text{tiff}}(x + u). \quad (3.7)$$

When u is odd, then $x + u$ is odd if x even, and is even otherwise. Using (3.6), $s_P(x)$ can be rewritten by

$$s_P^{\text{tiff}}(x) = \sum_{u \neq 0} k_u s_R^{\text{tiff}}(x + u) = \begin{cases} \sum_{u \neq 0} k_u \sum_{v \neq 0} h_v s_A^{\text{tiff}}(x + u + v) & \text{if } x \text{ even,} \\ \sum_{u \neq 0} k_u s_A^{\text{tiff}}(x + u) & \text{if } x \text{ odd.} \end{cases} \quad (3.8)$$

The signal residue, which is the difference between resulting and predictive signals, is thus expressed as

$$e^{\text{tiff}}(x) = s_R^{\text{tiff}}(x) - s_P^{\text{tiff}}(x) = \begin{cases} e_A^{\text{tiff}}(x) & \text{if } x \text{ even,} \\ e_I^{\text{tiff}}(x) & \text{if } x \text{ odd,} \end{cases} \quad (3.9)$$

where

$$e_A^{\text{tiff}}(x) = s_A^{\text{tiff}}(x) - \sum_{u \neq 0} k_u \sum_{v \neq 0} h_v s_A^{\text{tiff}}(x + u + v), \quad \text{for } x \text{ even,} \quad (3.10)$$

and

$$e_I^{\text{tiff}}(x) = \sum_{u \neq 0} h_u s_A^{\text{tiff}}(x + u) - \sum_{u \neq 0} k_u s_A^{\text{tiff}}(x + u) = \sum_{u \neq 0} (h_u - k_u) s_A^{\text{tiff}}(x + u), \quad \text{for } x \text{ odd.} \quad (3.11)$$

Since the size of kernel window is usually small, acquired signals in such a window can be assumed identical independent distributed (i.i.d.) with mean μ and variance σ^2 . Consequently, the mean $E[e^{\text{tiff}}(x)]$ and variance $\text{var}[e^{\text{tiff}}(x)]$ of the residue of TIFF image are obtained by Theorem 3.1.

Theorem 3.1. *If acquired signals of TIFF images are identically independently Gaussian distributed such that $s_A^{\text{tiff}}(x) \sim \mathcal{N}(\mu, \sigma^2)$, then the mean and variance of the residues e_A^{tiff} and e_I^{tiff} are expressed by*

- if x is even, then

$$E[e_A^{\text{tiff}}(x)] = 0 \quad \text{and} \quad \text{var}[e_A^{\text{tiff}}(x)] = \sigma^2 \cdot \left(1 + \sum_{u \neq 0} k_u^2 \sum_{v \neq 0} h_v^2\right), \quad (3.12)$$

- if x is odd, then

$$E[e_I^{\text{tiff}}(x)] = 0 \quad \text{and} \quad \text{var}[e_I^{\text{tiff}}(x)] = \sigma^2 \cdot \sum_{u \neq 0} (h_u - k_u)^2, \quad (3.13)$$

where k_u is an estimate of the interpolation kernel h_u .

Proof. We can derive the results in Theorem 3.1 from (3.10) and (3.11) as follows.

- If x is even, then using (3.10), we obtain

$$E [e_A^{\text{tiff}}(x)] = \mu - \sum_{u \neq 0} k_u \sum_{v \neq 0} h_v \mu = \mu \cdot \left(1 - \underbrace{\sum_{u \neq 0} k_u \sum_{v \neq 0} h_v}_{1} \right) = 0, \quad (3.14)$$

and

$$\begin{aligned} \text{var} [e_A^{\text{tiff}}(x)] &= \text{var} \left[s_A^{\text{tiff}}(x) - \sum_{u \neq 0} k_u \sum_{v \neq 0} h_v s_A^{\text{tiff}}(x + u + v) \right] = \underbrace{\text{var} [s_A^{\text{tiff}}(x)]}_{\sigma^2} + \\ &\quad \underbrace{\text{var} \left[\sum_{u \neq 0} k_u \sum_{v \neq 0} h_v s_A^{\text{tiff}}(x + u + v) \right]}_{0} - 2 \underbrace{\text{cov} \left[s_A^{\text{tiff}}(x), \sum_{u \neq 0} k_u \sum_{v \neq 0} h_v s_A^{\text{tiff}}(x + u + v) \right]}_{0}, \end{aligned} \quad (3.15)$$

where all terms of $\text{cov} [\cdot, \cdot]$ equal to 0 due to the independence of $s_A^{\text{tiff}}(x + u + v)$. Since

$$\begin{aligned} \text{var} \left[\sum_{u \neq 0} k_u \sum_{v \neq 0} h_v s_A^{\text{tiff}}(x + u + v) \right] &= \sum_{u \neq 0} k_u^2 \left(\text{var} \left[\sum_{v \neq 0} h_v s_A^{\text{tiff}}(x + u + v) \right] \right) = \\ &= \sum_{u \neq 0} k_u^2 \sum_{v \neq 0} h_v^2 \text{var} [s_A^{\text{tiff}}(x + u + v)] = \sum_{u \neq 0} k_u^2 \sum_{v \neq 0} h_v^2 \sigma^2 = \sigma^2 \cdot \sum_{u \neq 0} k_u^2 \sum_{v \neq 0} h_v^2, \end{aligned} \quad (3.16)$$

we can further express $\text{var} [e_A^{\text{tiff}}(x)]$ as

$$\text{var} [e_A^{\text{tiff}}(x)] = \sigma^2 \cdot \left(1 + \sum_{u \neq 0} k_u^2 \sum_{v \neq 0} h_v^2 \right). \quad (3.17)$$

- If x is odd, then we obtain from (3.11) that

$$E [e_I^{\text{tiff}}(x)] = \sum_{u \neq 0} h_u \mu - \sum_{u \neq 0} k_u \mu = \mu \cdot \left(\underbrace{\sum_{u \neq 0} h_u}_1 - \underbrace{\sum_{u \neq 0} k_u}_1 \right) = 0, \quad (3.18)$$

and

$$\begin{aligned} \text{var} [e_I^{\text{tiff}}(x)] &= \text{var} \left[\sum_{u \neq 0} (h_u - k_u) s_A^{\text{tiff}}(x + u) \right] \\ &= \sum_{u \neq 0} (h_u - k_u)^2 \text{var} [s_A^{\text{tiff}}(x + u)] = \sigma^2 \cdot \sum_{u \neq 0} (h_u - k_u)^2. \end{aligned} \quad (3.19)$$

The expressions (3.14), (3.17), (3.18) and (3.19) constitute the results of Theorem 3.1. \square

We find that $E[e^{\text{tiff}}(x)]$ is always equal to 0 for whatever position of x . Meanwhile, when the estimate of the interpolation kernel k_u is close to the original one h_u , $\text{var}[e^{\text{tiff}}(x)]$ is close to 0 at the positions of interpolated signal (i.e., $\text{var}[e_I^{\text{tiff}}(x)] \rightarrow 0$), while it is greater than σ^2 at the positions of acquired signal as $\sum_{u \neq 0} k_u^2 \sum_{v \neq 0} h_v^2 \in (0, 1]$ (i.e., $\sigma^2 < \text{var}[e_A^{\text{tiff}}(x)] \leq 2\sigma^2$). Therefore, the difference between variances of residues in acquired and interpolated signals can be seen as a useful demosaicing artifact for TIFF images.

3.2.3 Demosaicing Artifacts in JPEG Images

For storage, the high-quality TIFF image is compressed into JPEG format (see e.g., [196] for the detail of compression process). If a lossy JPEG compression is applied, the high-frequency components of 8×8 blocks in DCT domain are weakened by quantization which cannot be restored. This results in the local homogenization of 8×8 blocks in spatial domain. As in [134], we can model such a phenomenon by mixing the features of the acquired and interpolated signal residues of TIFF image. Let e_A^{tiff} and e_I^{tiff} be the representatives of acquired and interpolated signal residues in a certain row of 8×8 blocks of the TIFF image, the associated signal residues in 8×8 blocks of the JPEG image can be expressed by

$$\begin{cases} e_A^{\text{jpeg}} = \alpha \cdot e_A^{\text{tiff}} + (1 - \alpha) \cdot e_I^{\text{tiff}}, \\ e_I^{\text{jpeg}} = \alpha \cdot e_I^{\text{tiff}} + (1 - \alpha) \cdot e_A^{\text{tiff}}, \end{cases} \quad (3.20)$$

where $\alpha \in [0.5, 1]$ is a weighting factor related to the compression quality Q . When $\alpha = 0.5$, $e_A^{\text{jpeg}} = e_I^{\text{jpeg}} = \frac{1}{2}(e_A^{\text{tiff}} + e_I^{\text{tiff}})$; when $\alpha = 1$, $e_A^{\text{jpeg}} = e_A^{\text{tiff}}$ and $e_I^{\text{jpeg}} = e_I^{\text{tiff}}$. These two configurations correspond respectively to a very small value ($Q < 10$) and a very high value ($Q = 100$) of compression quality.

Let consider now the mean and variance of e_A^{jpeg} and e_I^{jpeg} . From (3.20), we derive, by using (3.12) and (3.13), that

$$E[e_A^{\text{jpeg}}] = E[e_I^{\text{jpeg}}] = 0, \quad (3.21)$$

and that

$$\begin{cases} \text{var}[e_A^{\text{jpeg}}] = \alpha^2 \cdot \text{var}[e_A^{\text{tiff}}] + (1 - \alpha)^2 \cdot \text{var}[e_I^{\text{tiff}}] + 2\alpha(1 - \alpha) \cdot \text{cov}[e_A^{\text{tiff}}, e_I^{\text{tiff}}], \\ \text{var}[e_I^{\text{jpeg}}] = \alpha^2 \cdot \text{var}[e_I^{\text{tiff}}] + (1 - \alpha)^2 \cdot \text{var}[e_A^{\text{tiff}}] + 2\alpha(1 - \alpha) \cdot \text{cov}[e_A^{\text{tiff}}, e_I^{\text{tiff}}]. \end{cases} \quad (3.22)$$

Taking the difference between $\text{var}[e_A^{\text{jpeg}}]$ and $\text{var}[e_I^{\text{jpeg}}]$, we obtain

$$\text{var}[e_A^{\text{jpeg}}] - \text{var}[e_I^{\text{jpeg}}] = (2\alpha - 1) \cdot (\text{var}[e_A^{\text{tiff}}] - \text{var}[e_I^{\text{tiff}}]). \quad (3.23)$$

Since $\text{var}[e_A^{\text{tiff}}] \geq \text{var}[e_I^{\text{tiff}}]$ and $\alpha \in [0.5, 1]$, we find that

$$0 \leq \text{var}[e_A^{\text{jpeg}}] - \text{var}[e_I^{\text{jpeg}}] \leq \text{var}[e_A^{\text{tiff}}] - \text{var}[e_I^{\text{tiff}}]. \quad (3.24)$$

As such, for the image in JPEG format, the difference between residues variances in acquired and interpolated signals is still a potential demosaicing artifact (i.e., $\text{var} [e_A^{\text{jpeg}}] \geq \text{var} [e_I^{\text{jpeg}}]$). However, as α decreases with respect to Q , we can easily derive from (3.22) that this difference becomes weaker when Q decreases, and is even disappeared when $Q < 10$ (for which $\alpha = 0.5$). This phenomenon has been numerically validated by Li *et al.* in [134] (see e.g., Figure 3 therein).

From the above analyses, we can conclude that, except some special configurations, the unbalance between the variances of signal residues at acquired and interpolated positions is an inherent property of digital images in TIFF and JPEG formats with high compression quality. When the image is strongly compressed, the unbalance is less clear. This explains in part why the demosaicing traces-based approach does not work well with JPEG compressed images.

3.3 Feature Statistics for Demosaiced Images

This section aims at extending the analysis in Section 3.2 to build feature statistics characterizing demosaiced images. To this end, we develop a complete process consisting of the identification of Bayer CFA pattern, the estimation of interpolation kernel, and the construction of feature statistics measuring the unbalance between the local variances of prediction residues. Especially, only the smooth region of green channel is considered to avoid perturbations due to edges. The main steps of the pro-

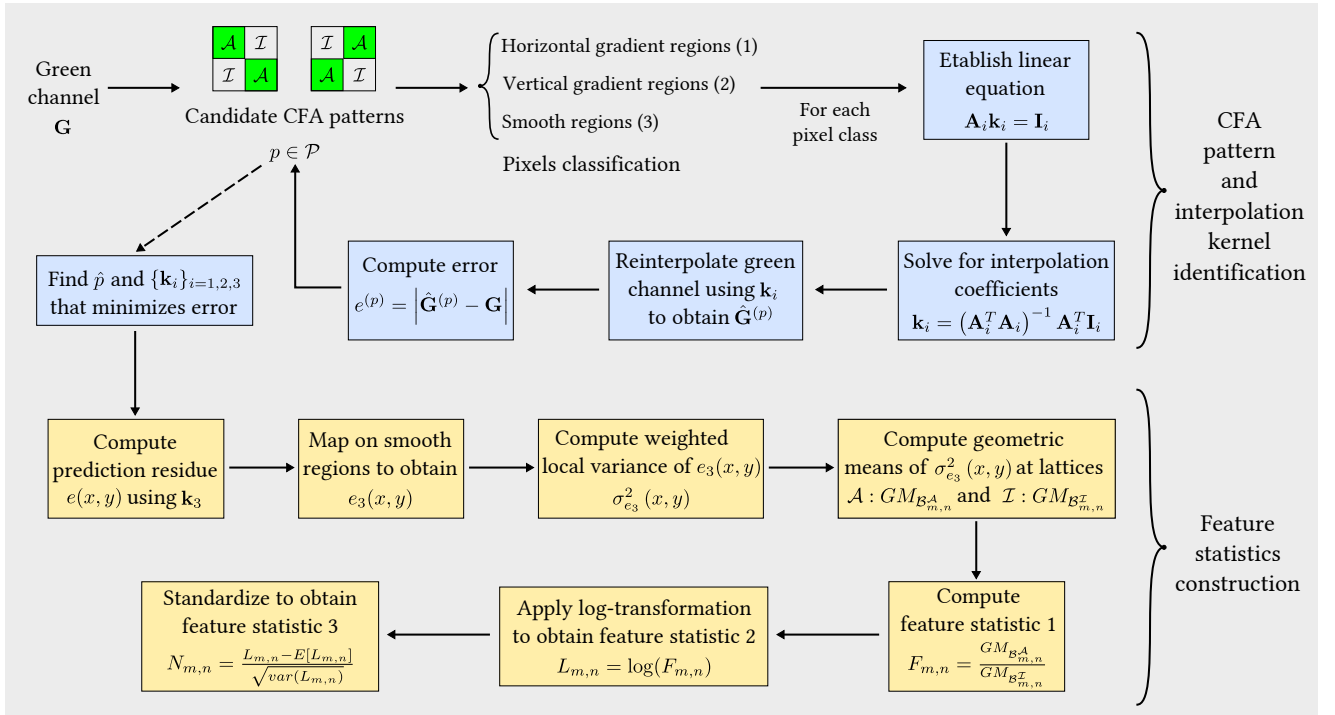


Figure 3.1: CFA pattern identification, interpolation kernel estimation, feature statistics construction

cess is schematically illustrated Figure 3.1. In the following, we provide the detailed developments for the process.

3.3.1 CFA Pattern and Interpolation Kernel

To avoid the edges effects, we adapt the identification scheme proposed by Swaminathan *et al.* in [199] to reveal the CFA pattern and interpolation kernel for the green channel of the considered image. With a given Bayer's CFA pattern $p \in \mathcal{P}$ (see Figure 3.1), we divide the green channel G into three kinds of regions \mathbf{R}_i , $i \in \{1, 2, 3\}$, based on the gradient features in a local neighborhood such that

$$\begin{cases} \mathbf{R}_1 = \{(x, y) \mid H(x, y) - V(x, y) \geq T\} & : \text{significant horizontal,} \\ \mathbf{R}_2 = \{(x, y) \mid V(x, y) - H(x, y) \geq T\} & : \text{significant vertical,} \\ \mathbf{R}_3 = \{(x, y) \mid |V(x, y) - H(x, y)| < T\} & : \text{mostly smooth,} \end{cases} \quad (3.25)$$

where T is a predetermined threshold, $H(x, y)$ and $V(x, y)$ are respectively the local gradient profile along the horizontal and vertical directions defined by

$$\begin{cases} H(x, y) = |s_R(x, y - 2) + s_R(x, y + 2) - 2s_R(x, y)| & \text{horizontal direction.} \\ V(x, y) = |s_R(x - 2, y) + s_R(x + 2, y) - 2s_R(x, y)| & \text{vertical direction,} \end{cases} \quad (3.26)$$

with $s_R(x, y)$ the resulting signal value at location (x, y) . For each region \mathbf{R}_i , we approximate interpolated signals with a set of linear equations of acquired signals, such that

$$\mathbf{A}_i \mathbf{k}_i = \mathbf{I}_i, \quad (3.27)$$

where \mathbf{A}_i , \mathbf{I}_i and \mathbf{k}_i , $i \in \{1, 2, 3\}$, are the matrix of acquired signals, the vector of interpolated signals, and the linear interpolation kernel of the region \mathbf{R}_i respectively. Solving (3.27) by the well-known least-squares method gives the following kernel \mathbf{k}_i

$$\mathbf{k}_i = \left(\mathbf{A}_i^T \mathbf{A}_i \right)^{-1} \mathbf{A}_i^T \mathbf{I}_i, \quad (3.28)$$

where \mathbf{A}_i^T and \mathbf{A}_i^{-1} denote respectively the transpose and inverse of matrix \mathbf{A}_i . Next, the obtained interpolation kernels are then used to reconstruct an estimation $\hat{G}^{(p)}$ of the green channel G . We repeat the above process to derive the reconstruction error for each CFA pattern. The optimal CFA pattern and interpolation kernel are jointly selected as the combination that yields the lowest reconstruction error.

3.3.2 Local Weighted Variance of Residues

The analysis in Section 3.2 is done under the assumption of i.i.d. acquired signals. To adapt its results to the two dimensional green channel of the digital image, we should evaluate the variance of prediction residues over small $(2K + 1) \times (2K + 1)$ windows, in which signal values are expected stationary. When sharp edges are present in the tested image, they may disturb this stationary property even for small windows. To overcome this obstacle, we propose using only the smooth region \mathbf{R}_3 in the computation

of the local variance of residues. By this way, the condition of i.i.d. acquired signals is guaranteed.

More precisely, we first compute the residue of the two dimensional green channel using the interpolated kernel \mathbf{k}_3 of the region \mathbf{R}_3

$$e(x, y) = s_R(x, y) - \sum_{u,v \neq 0} k_{3,u,v} s_A(x+u, y+v), \quad (3.29)$$

where $s_R(x, y)$ and $s_A(x, y)$ denote respectively the resulting and acquired signals at the location (x, y) of the green channel, and $k_{3,u,v}$ is an element in \mathbf{k}_3 . Then, mapping the residue $e(x, y)$ onto the smooth region R_3 , we obtain

$$e_3(x, y) = \begin{cases} e(x, y) & \text{if } (x, y) \in \mathbf{R}_3, \\ 0 & \text{otherwise.} \end{cases} \quad (3.30)$$

Following [26], the local weighted mean and local weighted variance of $e_3(x, y)$ within $(2K+1) \times (2K+1)$ windows can be respectively computed as

$$\mu_{e_3}(x, y) = \sum_{i,j=-K}^K \alpha_{i,j} e_3(x+i, y+j), \quad (3.31)$$

$$\sigma_{e_3}^2(x, y) = \frac{1}{c} \left(\sum_{i,j=-K}^K \alpha_{i,j} e_3^2(x+i, y+j) - \mu_{e_3}^2(x, y) \right), \quad (3.32)$$

where α_{ij} are suitable weights given by

$$\alpha_{ij} = \frac{\alpha'_{ij}}{\sum_{i,j} \alpha'_{ij}}, \quad (3.33)$$

in which

$$\alpha'_{ij} = \begin{cases} W(i, j) & \text{if } e_3(x+i, y+j) \text{ belong to the same class of } e_3(x, y), \\ 0 & \text{otherwise.} \end{cases} \quad (3.34)$$

$W(i, j)$ is a $(2K+1) \times (2K+1)$ Gaussian windows centered at (i, j) , and $c = 1 - \sum_{i,j=-K}^K \alpha_{ij}^2$ is a scale factor making the estimator unbiased such that $E[\sigma_{e_3}^2(x, y)] = \text{var}[\sigma_{e_3}(x, y)]$.

3.3.3 Definition and Efficiency of Feature Statistics

As shown in Section 3.2, the unbalance between residues variances of acquired and interpolated signals is inherent in demosaiced images. Here, the aim is to develop some feature statistics to expose this unbalance on the green channel locally. To this end, we first divide the variance map of residues with size $N \times N$ into $B \times B$ non-overlapping blocks, where B is a multiple of the length of Bayer's filter. Each $B \times B$ block is composed of $B^2/2$ acquired positions at quincunx lattices \mathcal{A} , and $B^2/2$ interpolated position at complementary quincunx lattices \mathcal{I} . Accordingly, we distinguish in a given (m, n) block $\mathcal{B}_{m,n}$, $m, n = 0, \dots, \frac{N}{B} - 1$, two sets: *acquired variances* $\mathcal{B}_{m,n}^{\mathcal{A}}$ and *interpolated variances* $\mathcal{B}_{m,n}^{\mathcal{I}}$. As in [63],

each of these sets can be characterized by the *geometric mean* of their elements

$$GM_{\mathcal{B}_{m,n}^{\mathcal{A}}} = \left(\prod_{m,n \in \mathcal{B}_{m,n}^{\mathcal{A}}} \sigma_{e_3}^2(m,n) \right)^{\frac{1}{B^2/2}} \quad \text{and} \quad GM_{\mathcal{B}_{m,n}^{\mathcal{I}}} = \left(\prod_{m,n \in \mathcal{B}_{m,n}^{\mathcal{I}}} \sigma_{e_3}^2(m,n) \right)^{\frac{1}{B^2/2}}. \quad (3.35)$$

Note that the *geometric mean* is used instead of the well-known *arithmetic mean* because it is less sensitive to extreme values. We can therefore define the unbalance between the local variance of signal residues at lattices \mathcal{A} and \mathcal{I} in the block $\mathcal{B}_{m,n}$ by the fraction

$$F_{m,n} = \frac{GM_{\mathcal{B}_{m,n}^{\mathcal{A}}}}{GM_{\mathcal{B}_{m,n}^{\mathcal{I}}}}. \quad (3.36)$$

By numerical experiments (see e.g., Figure 3.2b, 3.3b and 3.4b), we find that the probability density function (pdf) of $F_{m,n}$, $m, n = 0, \dots, \frac{N}{B} - 1$, is positively skewed with long tail on the right. Such a form does not allow feasible pdf fitting. This is why we apply the log-transformation to $F_{m,n}$ to favor the normality

$$L_{m,n} = \log GM_{\mathcal{B}_{m,n}^{\mathcal{A}}} - \log GM_{\mathcal{B}_{m,n}^{\mathcal{I}}} = \frac{1}{B^2/2} \sum_{m,n \in \mathcal{B}_{m,n}^{\mathcal{A}}} \log(\sigma_{e_3}^2(m,n)) - \frac{1}{B^2/2} \sum_{m,n \in \mathcal{B}_{m,n}^{\mathcal{I}}} \log(\sigma_{e_3}^2(m,n)). \quad (3.37)$$

Clearly, the feature $L_{m,n}$ represents the difference between the *arithmetic mean* of the logarithm of variances in the set $\mathcal{B}_{m,n}^{\mathcal{A}}$ and $\mathcal{B}_{m,n}^{\mathcal{I}}$. The mean and variance of $L_{m,n}$ vary from image to image. To have a feature statistic independent of image content, $L_{m,n}$ is standardized

$$N_{m,n} = \frac{L_{m,n} - E[L_{m,n}]}{\sqrt{\text{var}[L_{m,n}]}} \quad (3.38)$$

so that the mean and variance of $N_{m,n}$ are fixed to 0 and 1 respectively.

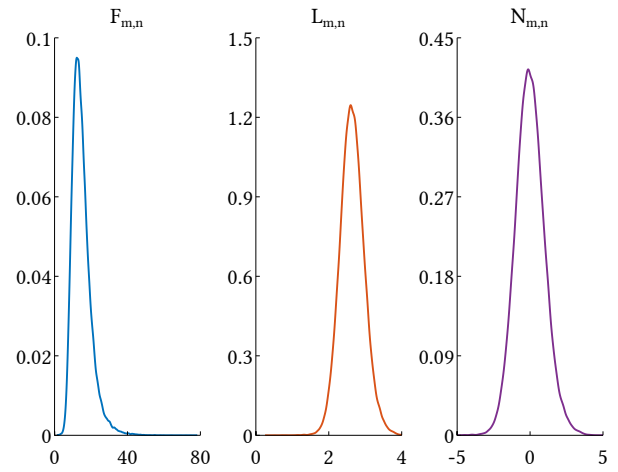
Since all kind of signals (i.e., the entire pixel, its content part, or its noise part) could be used to compute the feature statistics $F_{m,n}$, $L_{m,n}$ and $N_{m,n}$, two questions arise naturally.

1. Which one among the three feature statistics $F_{m,n}$, $L_{m,n}$ and $N_{m,n}$ is more relevant to characterize demosaiced images?
2. Which kinds of signals allow the most robust feature statistic?

The following numerical experiments allow to answer to these questions. Let consider the uncompressed color image Berries.png taken from the Image Manipulation Dataset [41], its green channel, the content part and the noise part are respectively displayed in Figures 3.2a, 3.3a and 3.4a. For each kind of signals, we compute the local weighted variance map of prediction residues (with $K = 3$), and derive the three feature statistics of interest $F_{m,n}$, $L_{m,n}$ and $N_{m,n}$ (with $B = 8$). Next, we show in Figures 3.2b, 3.3b and 3.4b the pdf of $F_{m,n}$, $L_{m,n}$ and $N_{m,n}$ returned by the *kernel density estimation* (KDE) method [23]. Clearly, the pdf forms of $F_{m,n}$ are uncontrollable for all kinds of signals. For the



(a) Green channel

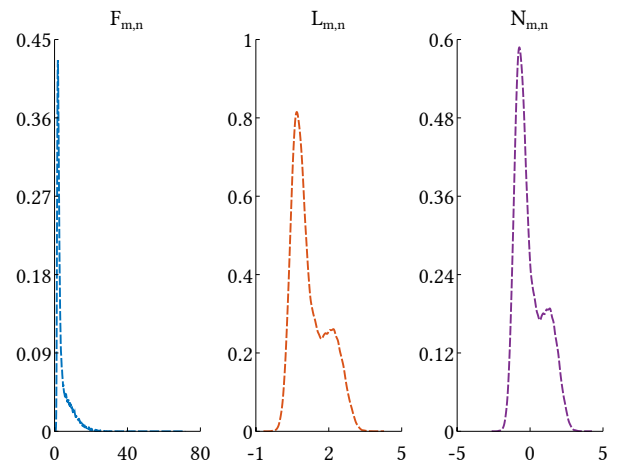


(b) Pdf of $F_{m,n}$, $L_{m,n}$ and $N_{m,n}$

Figure 3.2: Green channel and pdf of feature statistics

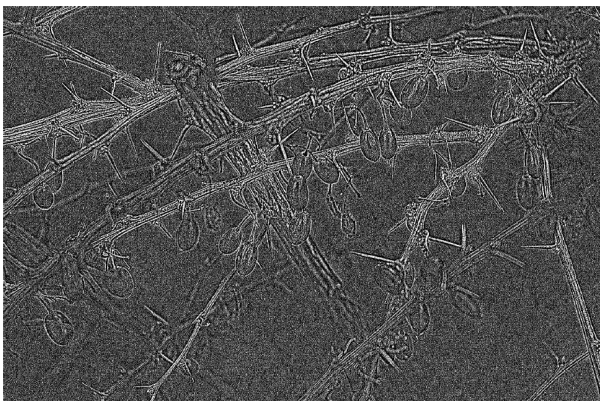


(a) Content part

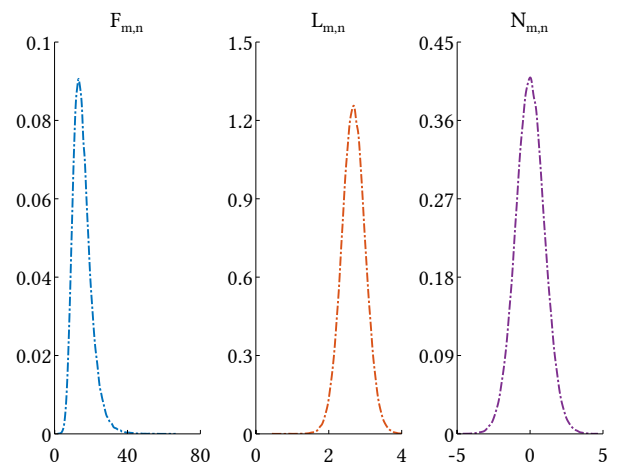


(b) Pdf of $F_{m,n}$, $L_{m,n}$ and $N_{m,n}$

Figure 3.3: Content part and pdf of feature statistics



(a) Noise part



(b) Pdf of $F_{m,n}$, $L_{m,n}$ and $N_{m,n}$

Figure 3.4: Noise part and pdf of feature statistics

pdf forms of $L_{m,n}$ and $N_{m,n}$, the ones obtained from the green channel or its noise part have a nice bell shape as expected, while the ones computed from the content part have not. The reason is that the weak demosaicing traces in the content part is easily covered and/or strongly affected by the real scene data. Meanwhile, for the noise part, although the demosaicing traces is still weak, the energy of noise is relatively small. So that the portion of demosaicing traces in noise is remarkable. The energy of pixels in green channel is high, but demosaicing traces therein are also much stronger. Consequently, the demosaicing artifacts given from the green channel are still significant. Compared to $L_{m,n}$, the advantage of $N_{m,n}$ is that its pdf is independent of image content, and hence more suitable for detection goal. To find out which one between the green channel and its noise part can provide robust $N_{m,n}$, we vary the size of $B \times B$ blocks and sketch the associated pdf of $N_{m,n}$ in Figure 3.5. Clearly, the pdf of $N_{m,n}$ given from the noise part is less sensitive to the value of B (especially since

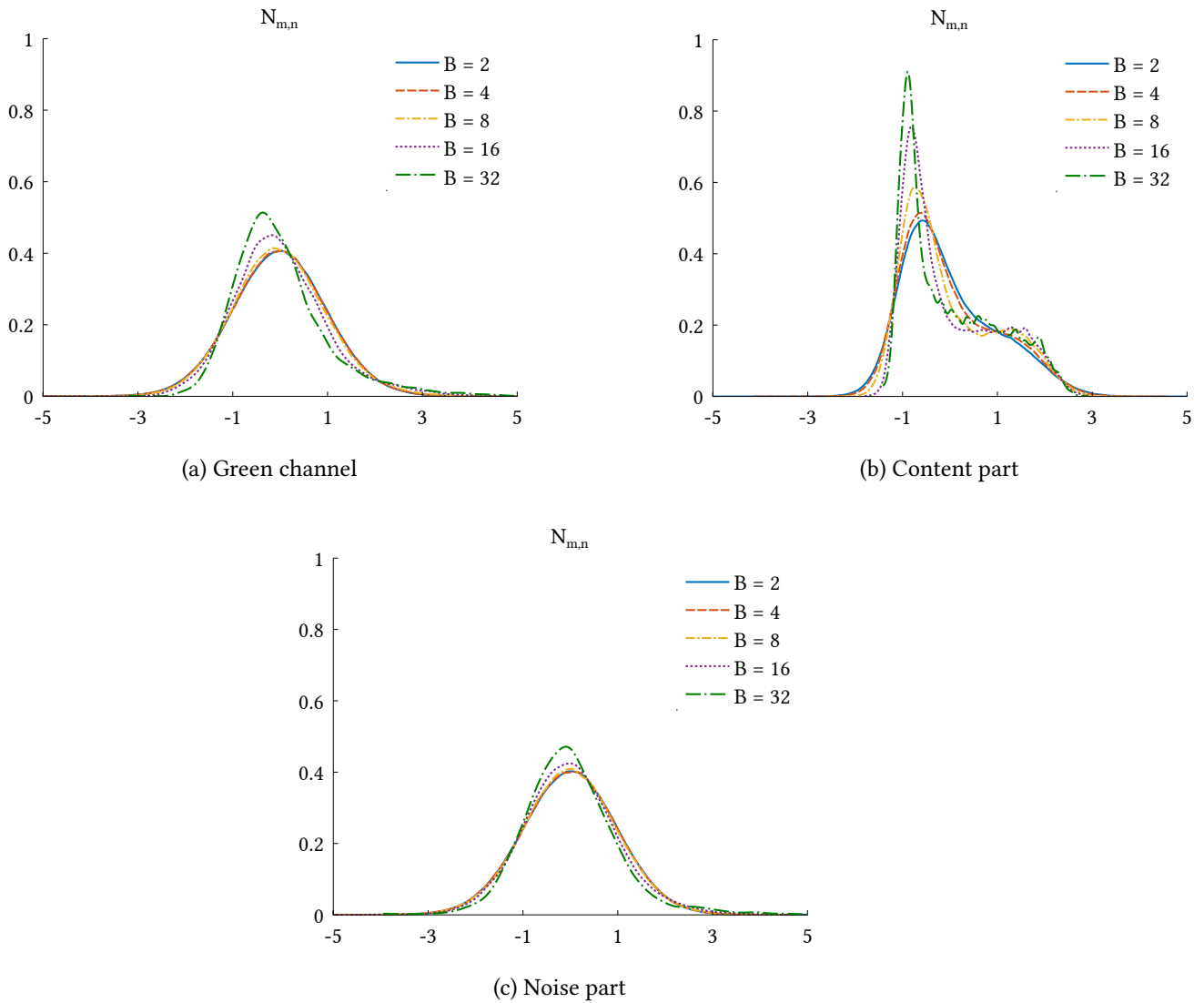


Figure 3.5: pdf of $N_{m,n}$ with respect to B

$B = 8$). Similar results given by repeating this experiment for various images allow us to confirm that the feature statistic $N_{m,n}$ computed from the noise part is robustest.

Hereafter, except clearly stated, we use $N_{m,n}$ computed from the noise part as the default feature statistic. Besides, to simplify the notation, we use a single index instead of double of indices to indicate the feature statistic (i.e., using N_i to denote the feature statistic associated with the i -th $B \times B$ block).

3.4 Image Authentication and Forgery Localization

This section aims to develop a single statistical model based on the feature statistic N_i for both natural and tampered digital images. Such a model is next used to authenticate digital images and to localize the tampered regions if any.

3.4.1 Statistical Model of Feature Statistic

As mentioned in Subsection 3.3.3, when the tested image is authentic, N_i is consistent and distributed following a standard normal distribution $\mathcal{N}(0, 1)$. When some regions in the image have been manipulated by a new content coming either from other regions of the same image (e.g., copy-move forgery, inpainting forgery), or from another images (e.g., splicing forgery), the demosaicing traces in these regions are normally different than the remainder of the image. The feature statistic N_i of such a tampered image is no longer consistent. In this case, we expect that values of N_i come from two different populations \mathcal{P}_1 and \mathcal{P}_2 corresponding to the untampered and tampered regions respectively. Each region is part of demosaiced images, so \mathcal{P}_k , $k \in \{1, 2\}$, is a normal distribution with mean μ_k and variance σ_k . For simplicity, we also set $\sigma_1 = \sigma_2 = \sigma$, because the mean parameters contribute to the separation between \mathcal{P}_1 and \mathcal{P}_2 in most situations. As a result, the population of N_i for an entire tampered image can be modeled as a normal mixture distribution $\mathcal{NM}(\gamma, \mu_1, \mu_2, \sigma)$ with pdf

$$f(N_i; \gamma, \mu_1, \mu_2, \sigma) = (1 - \gamma) \cdot f(N_i; \mu_1, \sigma) + \gamma \cdot f(N_i; \mu_2, \sigma), \quad (3.39)$$

where γ , $0 \leq \gamma \leq 1$, denotes the proportion of the population 2, and $f(x; \mu_k, \sigma)$ standards for the normal p.d.f $\mathcal{N}(\mu_k, \sigma)$

$$f(N_i; \mu_k, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(N_i - \mu_k)^2}. \quad (3.40)$$

As in [183], to avoid the nonidentifiability of (3.39), we set $0 \leq \gamma \leq 0.5$. This implies that the population 2 of the model (3.39) is the tampered population \mathcal{P}_2 under the assumption that tampered regions are smaller than the remainder of the image. When $\mu_1 = \mu_2$ or $\gamma = 0$, the model (3.39) degenerates into a single normal distribution. By this way, we can also use (3.39) as a model for both authentic and tampered images.

3.4.2 Estimation of Model Parameters

Given the model (3.39), the next issue is to estimate the parameters γ , μ_1 , μ_2 and σ from the set of feature statistics N_i . A penalized EM algorithm has been developed for this issue. Let $\mathbf{N} = \{N_1, \dots, N_q\}$ be a set of q feature statistics sampled from a normal mixture population $\mathcal{NM}(\gamma, \mu_1, \mu_2, \sigma)$, its ordinary

log-likelihood function is given by

$$l_q(\gamma, \mu_1, \mu_2, \sigma) = \sum_{i=1}^q \log((1-\gamma) \cdot f(N_i; \mu_1, \sigma) + \gamma \cdot f(N_i; \mu_2, \sigma)). \quad (3.41)$$

As proved in [31], $l_q(\gamma, \mu_1, \mu_2, \sigma) \rightarrow \infty$ if $\mu_k \rightarrow N_i$ and $\sigma \rightarrow 0$ with the other parameters fixed. This implies that ordinary maximum-likelihood estimator of $(\gamma, \mu_1, \mu_2, \sigma)$ is not well-defined [50, 104]. To remedy, Chen *et al.* [31] propose adding penalty term to the ordinary log-likelihood function. Such an approach has been proved efficient because of the strong consistency of maximum likelihood estimators for various penalties on σ [31, 43]. Therefore, we can define a penalized log-likelihood function as

$$pl_q(\gamma, \mu_1, \mu_2, \sigma) = l_q(\gamma, \mu_1, \mu_2, \sigma) + p_q(\sigma), \quad (3.42)$$

where $p_q(\sigma)$ is the penalty function on σ . To compensate the aforementioned undesirable configuration, we should select $p_q(\sigma)$ such that it is bounded when σ is large, but goes to $-\infty$ as $\sigma \rightarrow 0$. Chen *et al.* [32] have recommended

$$p_q(\sigma) = -a_q \cdot \left(\frac{s_q^2}{\sigma^2} + \log\left(\frac{\sigma^2}{s_q^2}\right) \right), \quad (3.43)$$

where $s_q^2 = \frac{1}{q} \sum_{i=1}^q N_i^2$ denotes the sample variance (the sample mean $\bar{N} = \frac{1}{q} \sum_{i=1}^q N_i$ is always 0), and a_q is a positive tuning parameter. A large value of a_q implies a strong conviction in the prior estimate of σ [31]. As in the works [33, 97], we choose $a_q = \frac{1}{q}$.

Until now, the estimation problem returns to find the tuple $(\hat{\gamma}, \hat{\mu}_1, \hat{\mu}_2, \hat{\sigma})$ that maximizes penalized log-likelihood function (3.42). Moreover, we would like to perform a population clustering for the set of feature statistics \mathbf{N} . This is why we try to introduce a vector of binary latent variables $\mathbf{Z} = \{Z_1, \dots, Z_q\}$ with $Z_i = (Z_{i1}, Z_{i2})$ indicating the cluster of a sample N_i in \mathbf{N} . The variable Z_{ik} , $k \in \{1, 2\}$, is defined as follows

$$Z_{ik} = \begin{cases} 1 & \text{if } N_i \text{ is from the population } k \\ 0 & \text{otherwise} \end{cases}, \quad (3.44)$$

and $\sum_{k=1}^2 Z_{ik} = 1$. The maximizing the likelihood estimation is now done on the complete data $\{\mathbf{N}, \mathbf{Z}\}$ rather than on the incomplete data only \mathbf{N} . As pointed out in [21, page 431], the joint pdfs of Z_i and $N_i | Z_i$ are respectively

$$f(Z_i) = \prod_{k=1}^2 \gamma_k^{Z_{ik}}, \quad (3.45)$$

and

$$f(N_i | Z_i) = \prod_{i=1}^q f^{Z_{ik}}(N_i; \mu_k, \sigma), \quad (3.46)$$

where $\gamma_1 = 1 - \gamma$, $\gamma_2 = \gamma$, and $f(N_i; \mu_k, \sigma)$ is given from (3.40). Therefore, the complete data likelihood

function takes the form

$$f(\mathbf{N}, \mathbf{Z}) = \prod_{i=1}^q f(N_i, Z_i) = \prod_{i=1}^q f(N_i | Z_i) f(Z_i) = \prod_{i=1}^q \prod_{k=1}^2 \gamma_k^{Z_{ik}} f^{Z_{ik}}(N_i; \mu_k, \sigma). \quad (3.47)$$

Taking the logarithm, we obtain the complete data log-likelihood function

$$l_c(\gamma, \mu_1, \mu_2, \sigma) = \sum_{i=1}^q \sum_{k=1}^2 Z_{ik} \cdot \left(\log \gamma_k - \log \sigma - \frac{1}{2\sigma^2} (x - \mu_k)^2 - \frac{1}{2} \log 2\pi \right). \quad (3.48)$$

Adding $p_q(\sigma)$, we obtain the complete data penalized log-likelihood function

$$pl_c(\gamma, \mu_1, \mu_2, \sigma) = l_c(\gamma, \mu_1, \mu_2, \sigma) + p_q(\sigma). \quad (3.49)$$

We find that $pl_c(\gamma, \mu_1, \mu_2, \sigma)$ can be trivially maximized in closed form. Unfortunately, we do not have values for the latent variables \mathbf{Z} , thus we cannot use directly $pl_c(\gamma, \mu_1, \mu_2, \sigma)$. Instead, we consider firstly its expected value under the posterior distribution of the latent variables (i.e., E-step of the EM algorithm). Next, we maximize this expectation (i.e., M-step of the EM algorithm). Such a procedure can be summarized by Algorithm 3.1 (see also Appendix A for the detailed formalization of the penalized EM algorithm).

3.4.3 Image Authentication and Forgery Localization

The above analyses allow to model the image authentication as a decision problem between two hypotheses

$$\begin{cases} \mathcal{H}_0 : N_i \sim \mathcal{N}(0, 1) & \text{authentic image,} \\ \mathcal{H}_1 : N_i \not\sim \mathcal{N}(0, 1) & \text{no conclusion.} \end{cases} \quad (3.57)$$

In reality, dazzling areas or missing color ranges existing on digital images may distort the distribution of N_i . Therefore, the distribution of N_i does not always have a perfect Gaussian form even for authentic images. To partially weaken the distortion impacts, we propose regenerating random samples \hat{N}_i from the model (3.39) using the estimated parameters obtained by Algorithm 3.1 (see e.g., [118, page 53] for a simulation algorithm). Then, we work with \hat{N}_i instead of N_i . As such, the problem (3.57) becomes

$$\begin{cases} \mathcal{H}_0 : \hat{N}_i \sim \mathcal{N}(0, 1) & \text{authentic image,} \\ \mathcal{H}_1 : \hat{N}_i \not\sim \mathcal{N}(0, 1) & \text{no conclusion.} \end{cases} \quad (3.58)$$

Normality tests are next carried out to decide if the tested image is authentic or not. In this chapter, well-known normality tests [224] (i.e., Anderson-Darling test, one-sample Kolmogorov-Smirnov test, Jarque-Bera test and Lilliefors test) are jointly used to achieve a reliable decision. If all these tests return \mathcal{H}_0 , the image is decided to be authentic automatically. Additionally, the authentication can be done by human interpretation thanks to graphical tools such Q-Q plot, probability distribution curves,

Algorithm 3.1 Penalized EM algorithm for parameters estimation and clustering

1. **I-step:** For $k = 1, 2$, get $\gamma^{(0)}$ from the k -means algorithm [21, chapter 9], then cluster \mathbf{N} , and compute $\mu_k^{(0)}$ and $\sigma_k^{(0)}$ as the maximum likelihood estimates of $\mathcal{N}(\mu_k, \sigma_k)$

$$\mu_k^{(0)} = \bar{N}_k = \frac{1}{q_k} \sum_{i=1}^{q_k} N_i, \quad \text{and} \quad \sigma_k^{(0)} = s_n = \sqrt{\frac{1}{q_k} \sum_{i=1}^{q_k} (N_i - \bar{N}_k)^2}, \quad (3.50)$$

where q_k is the number of elements of the population k . Set $\sigma^{(0)} = \sigma_k^{(0)}$ if k is the larger population.

2. **E-step:** Given \mathbf{N} and the vector of parameters $\Theta^{(t)} = (\gamma^{(t)}, \mu_1^{(t)}, \mu_2^{(t)}, \sigma^{(t)})$ at the t -th current iteration, compute

$$\gamma_{i1}^{(t+1)} = \frac{(1 - \gamma^{(t)}) f(N_i; \mu_1^{(t)}, \sigma^{(t)})}{(1 - \gamma^{(t)}) f(N_i; \mu_1^{(t)}, \sigma^{(t)}) + \gamma^{(t)} f(N_i; \mu_2^{(t)}, \sigma^{(t)})}, \quad (3.51)$$

and

$$\gamma_{i2}^{(t+1)} = \frac{\gamma^{(t)} f(N_i; \mu_2^{(t)}, \sigma^{(t)})}{(1 - \gamma^{(t)}) f(N_i; \mu_1^{(t)}, \sigma^{(t)}) + \gamma^{(t)} f(N_i; \mu_2^{(t)}, \sigma^{(t)})}. \quad (3.52)$$

3. **M-step:** Update the set of parameter $\Theta^{(t)}$ as follows.

- (a) Update $\gamma^{(t)}$ by

$$\gamma^{(t+1)} = \frac{\sum_{i=1}^q \gamma_{i2}^{(t+1)}}{q}. \quad (3.53)$$

- (b) Update $\mu_k^{(t)}$, $k \in \{1, 2\}$ by

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^q \gamma_{ik}^{(t+1)} N_i}{\sum_{i=1}^q \gamma_{ik}^{(t+1)}}. \quad (3.54)$$

- (c) Update $\sigma^{(t)}$ by

$$\sigma^{(t+1)} = \sqrt{\frac{S^{(t+1)} + 2a_q s_q^2}{q + 2a_q}}. \quad (3.55)$$

where

$$S^{(t+1)} = \sum_{i=1}^q \sum_{k=1}^2 \gamma_{ik}^{(t+1)} (N_i - \mu_k^{(t+1)})^2. \quad (3.56)$$

4. **S-step:** Stop the EM algorithm whenever the iteration number t exceeds a limited number, or when the norm $\|\Theta^{(t+1)} - \Theta^{(t)}\|$ is smaller a threshold. Set $(\hat{\gamma}, \hat{\mu}_1, \hat{\mu}_2, \hat{\sigma})$ as the final $\Theta^{(t+1)}$, and stock the final $\gamma_{i1}^{(t+1)}$ and $\gamma_{i2}^{(t+1)}$ for feature statistics clustering.

and localization map.

Regarding the forgery localization, we can decide the belonging of N_i to \mathcal{P}_1 and \mathcal{P}_2 from the set $\{\gamma_{i1}\}_{i=1,\dots,q}$ and $\{\gamma_{i2}\}_{i=1,\dots,q}$ (see the output of Algorithm 3.1) as follow

$$N_i \in \begin{cases} \mathcal{P}_1 : \text{authentic portion} & \text{if } \gamma_{i1} > \gamma_{i2} \\ \mathcal{P}_2 : \text{tampered portion} & \text{if } \gamma_{i1} \leq \gamma_{i2} \end{cases}, \quad i = 1, \dots, q. \quad (3.59)$$

Since each N_i correspond to the i -th $B \times B$ block of the tested image, we can therefore derive a binary map indicating tampered region of the image.

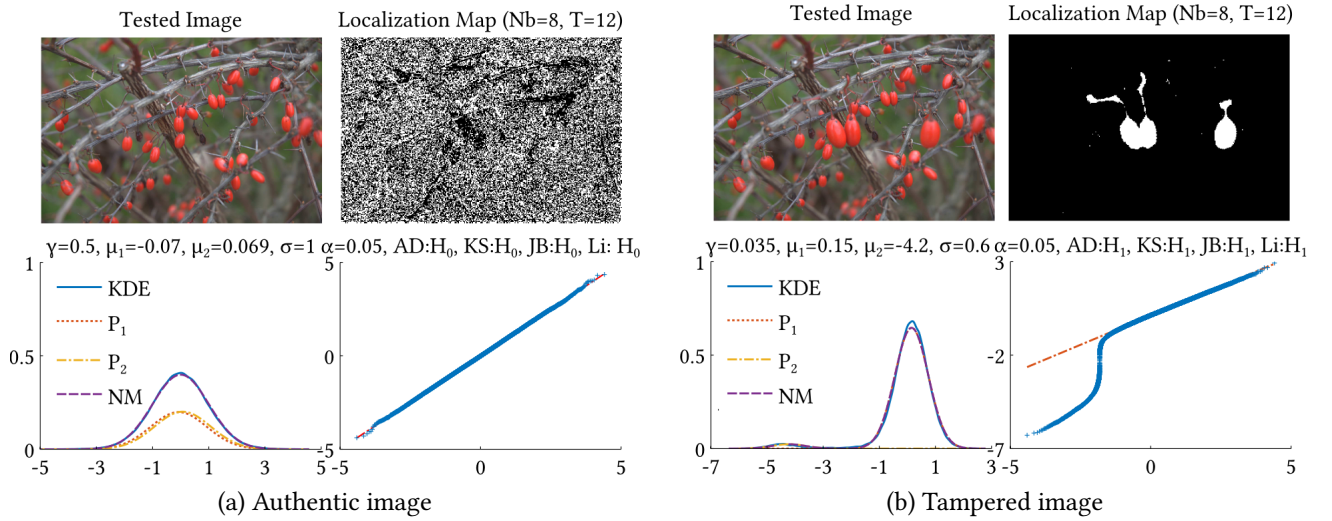


Figure 3.6: Authentication and localization results for Berries.png images

For an illustration, we show in Figures 3.6a and 3.6b the authentication and localization results for an authentic image and a tampered image respectively. For the authentic image, the distribution of feature statistics is standard normal. Accordingly, with a type 1 error $\alpha = 0.05$, all the considered normality tests return \mathcal{H}_0 , while the points in Q-Q plot diagram follow a linear pattern. This confirms the authenticity of the image. Looking at the localization map, black and white blocks are equally mixed overall the image because $\gamma = 0.5$, and no concrete form is appeared. For the tampered image, the distribution of feature statistics is no longer standard normal. All the normality tests return then \mathcal{H}_1 , and the points in Q-Q plot diagram follow a strongly nonlinear pattern. The outliers of dash orange line in Q-Q plot diagram correspond to the smaller population in the mixture pdf of feature statistics. The location map now shows clearly the tampered regions in white and authentic region in black.

3.5 Numerical Experiments

Image Manipulation Dataset [41], MICC-F600 Dataset [7], Realistic Tampering Dataset [115], and Columbia Uncompressed Image Splicing Detection Evaluation Dataset [89] are used for numerical

experiments. Their detailed description can be found in [114]. Through these datasets, we aim at evaluating the images authentication and forgeries localization ability of the proposed algorithm.

3.5.1 Evaluation Criteria and Benchmark Algorithms

Due to highly imbalanced datasets¹, Precision (P), Recall (R) and F_1 -Score (F_1) are chosen as criteria for performance evaluation [5]. Precision and Recall are computed from the confusion matrix of True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) as

$$P = \frac{TP}{TP + FP} \quad \text{and} \quad R = \frac{TP}{TP + FN}. \quad (3.60)$$

A high value of P implies a high probability that detected results are relevant, while a high value of R means a high probability that relevant results are detected. Obviously, using separately either Precision or Recall is not enough to evaluate the performance of an algorithm. The harmonic mean of Precision and Recall, called F_1 -Score, might be a better measure as it takes a high value when Precision and Recall are both important

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R} = \frac{2 \cdot TP}{2 \cdot TP + FN + FP}. \quad (3.61)$$

The higher F_1 -Score, the more the algorithm is efficient. Moreover, depending on the authentication or localization goals, the above measures has their own meanings which are detailed in Table 3.1.

	Images authentication (image level)	Forgery localization (pixel level)
TP	number of authentic images correctly detected	number of tampered pixels correctly localized
FN	number of undetected authentic images	number of unlocalized tampered pixels
FP	number of tampered images wrongly detected	number of authentic pixels wrongly localized
TN	number of undetected tampered images	number of unlocalized authentic pixels
P	probability that detected images are authentic	probability that localized pixels are tampered
R	probability that authentic images are detected	probability that tampered pixels are localized

Table 3.1: Meaning of performances measures

The performance and robustness of the developed algorithm are assessed through comparative studies with algorithms proposed by Dirik and Menon in [55], and by Ferrara *et al.* in [63]. These two benchmarks are threshold-based algorithms, so their performance depend closely on a threshold τ used to distinguish between tampered and authentic regions in an image. Meanwhile, the performance of the developed algorithm is dependent on the threshold T used to determine the smooth region \mathbf{R}_3 . Besides, the blocks size B is influential in the performance of all the algorithms. So, sensitivity studies to these factors are necessary. We also note that the two benchmark algorithms focuses on images forgery localization rather than on images authentication.

¹Datasets here refer to authentic and tampered images in the case of image authentication, and to untampered and tampered pixels in the case of forgery localization

3.5.2 Authentication Performance

To assess the authentication performance of the proposed algorithm, we first randomly choose a set of 150 authentic images and 200 tampered images from the 4 above datasets. Next, we apply the algorithm to obtain TP , FN , FP and TN , and thence compute the measures P , R and F_1 according to (3.60) and (3.61). Such a process is applied to the 2 following sensitivity studies.

- *Sensitivity study 1*: B takes respectively the value 2, 4, 8, 16 and 32, while T is fixed at 20.
- *Sensitivity study 2*: B is fixed at 16, and T varies from 10 to 30 with step 5.

Table 3.2 presents the results when the authentication is done automatically by the proposed method. Clearly, the performance of the automatic authentication is relatively weak. Indeed, despite working

Case study 1 ($T = 20$)						Case study 2 ($B = 16$)					
B	2	4	8	16	32	T	10	15	20	25	30
P (%)	46.02	42.59	37.50	45.59	57.63	P (%)	45.33	43.94	45.59	43.42	39.39
R (%)	34.67	30.67	22.00	20.67	22.67	R (%)	22.67	19.33	20.67	22.00	17.33
F_1 (%)	39.54	35.66	27.73	28.44	32.54	F_1 (%)	30.22	26.85	28.44	29.20	24.07

Table 3.2: Results of automatic authentication

on the smooth region R_3 instead of on the entire green channel G , and using \hat{N}_i rather than N_i , shape edges and/or strong dazzling areas existing on realistically authentic digital images still distort the standard Gaussian form of \hat{N}_i pdf. As a results, the automatic authentication via normality tests is less efficient. However, as shown in Table 3.3, the performance is much more improved thanks to human interpretation of graphic tools (i.e., pdf curves, Q-Q plot, and localization map). To better

Case study 1 ($T = 20$)						Case study 2 ($B = 16$)					
B	2	4	8	16	32	T	10	15	20	25	30
P (%)	59.82	59.45	64.67	61.59	63.64	P (%)	63.80	64.15	61.59	62.58	59.73
R (%)	87.33	86.00	79.33	62.00	46.67	R (%)	69.33	68.00	62.00	64.67	59.33
F_1 (%)	71.00	70.30	71.26	61.79	53.85	F_1 (%)	66.45	66.02	61.79	63.61	59.53

Table 3.3: Authentication results given by human interpretation

see how the human interpretation can help, let us introduce an example as in Figure 3.7. The tested images in Figures 3.7a and 3.7b are respectively authentic and tampered. If only the results returned by normality tests are taken into account (i.e., automatic authentication), “no conclusion” is decided for the first image, while “authentic image” is decided for the second one. They are obviously wrong decisions. Nevertheless, if we look more at the pdf curves, the Q-Q plot, and mostly at the localization map, it is not hard to decide that the first and second tested images are authentic and tampered respectively, which is actually true. As such, the human interpretation can help to improve the performance of image authentication.

Looking at the values of F_1 -Score in Tables 3.2 and 3.3, we also find the importance of B and T . The increasing of blocks size B leads to smaller samples set in constructing pdf of \hat{N}_i . Meanwhile, by

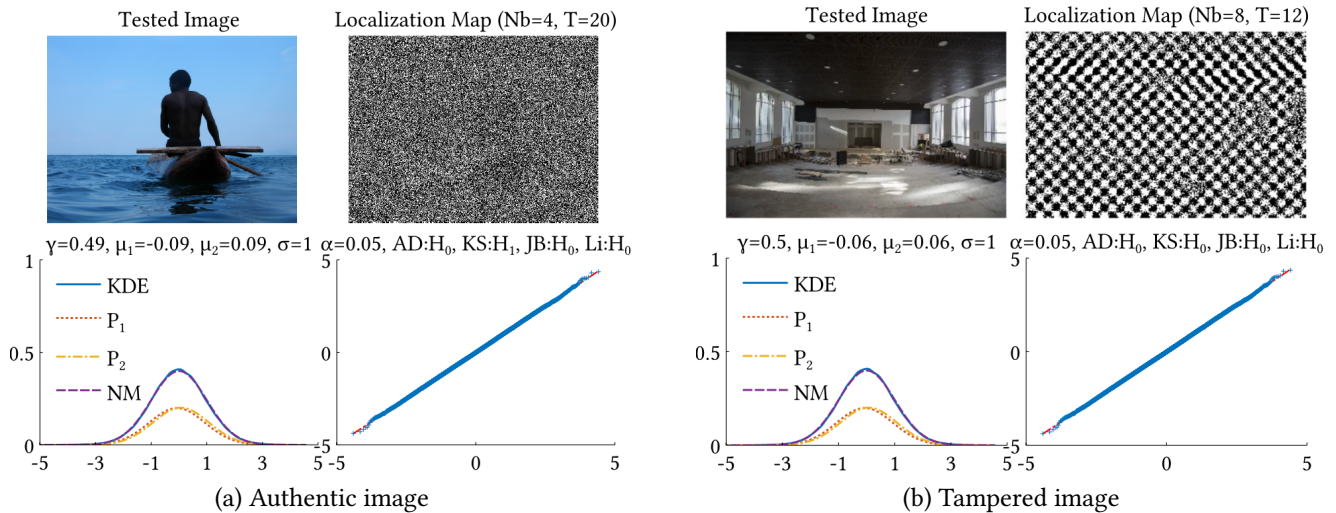


Figure 3.7: Authentication and localization results for digital images

setting T at a high value, more edges are allowed in the computation of N_i . So, it is not surprising that the authentication performance decreases in these configurations of T and B .

3.5.3 Localization Performance

Regarding the localization, the aim is to find out (i) which kinds of forgeries could be localized by the proposed algorithm, and (ii) how good is the proposed algorithm compared to algorithms of the same kind. Tampered images in Image Manipulation Dataset [41] and Realistic Tampering Dataset [115] are used in numerical experiments because of various kinds of forgeries therein. Moreover, binary ground truths are also provided.

To reply to the first issue, we use the proposed algorithm (with $T = 20$ and blocks size $B = 8$) to derive location maps, and thence compare them with the associated ground truths. We also apply the algorithms proposed by Dirik and Menon [55], and by Ferrara *et al.* [63] to obtain benchmark location maps. As illustrated in Figure 3.8, various kinds of forgeries are successfully localized by our algorithm. Clearly, localization maps returned by our algorithm and Ferrara *et al.*'s algorithm are closer to ground truths, and more confident than Dirik and Menon's algorithm. When zooming in the localized areas, our algorithm allows better resolution and higher fidelity than the Ferrara *et al.*'s algorithm. We will understand more deeply the reason through a quantitative study for the second issue in the following. Moreover, the use of Ferrara *et al.*'s algorithm is more complicated in practice because a subjective threshold is required to distinguish tampered and authentic region. Figure 3.9 shows some configurations that forgeries are not successfully localized. For cloning forgery (first line of Figure 3.9), a part of tampered region is missing because it has the same alignment of CFA pattern as the original image. Especially, the localization is completely failed (see second line and third line of Figure 3.9), if tampered images undergo additional post-processing (e.g., JPEG compression with low quality, down-sampling, etc.).

To reply to the second issue, we first apply the 3 considered algorithms to compute localization

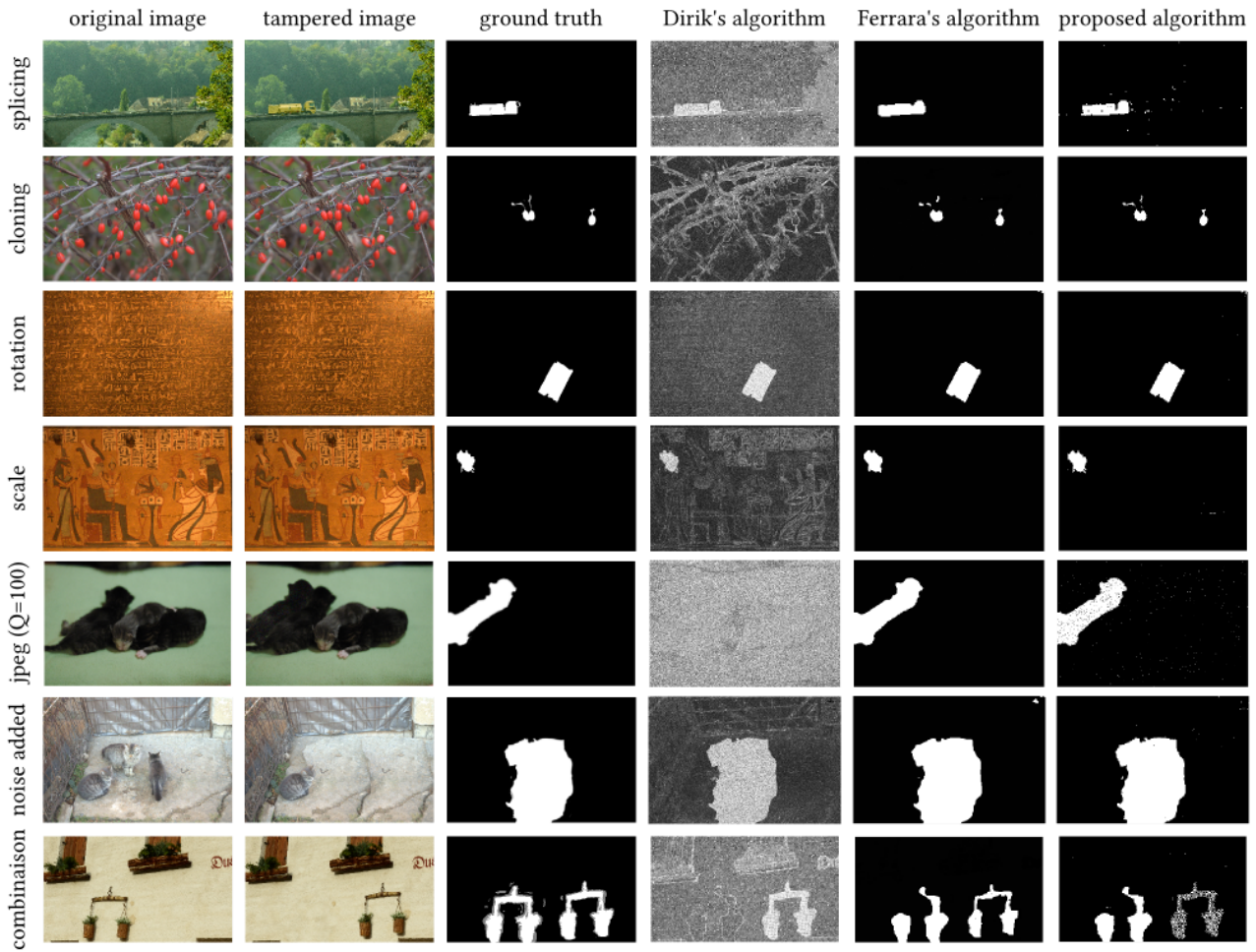


Figure 3.8: Examples of successful forgery localization by the proposed algorithm

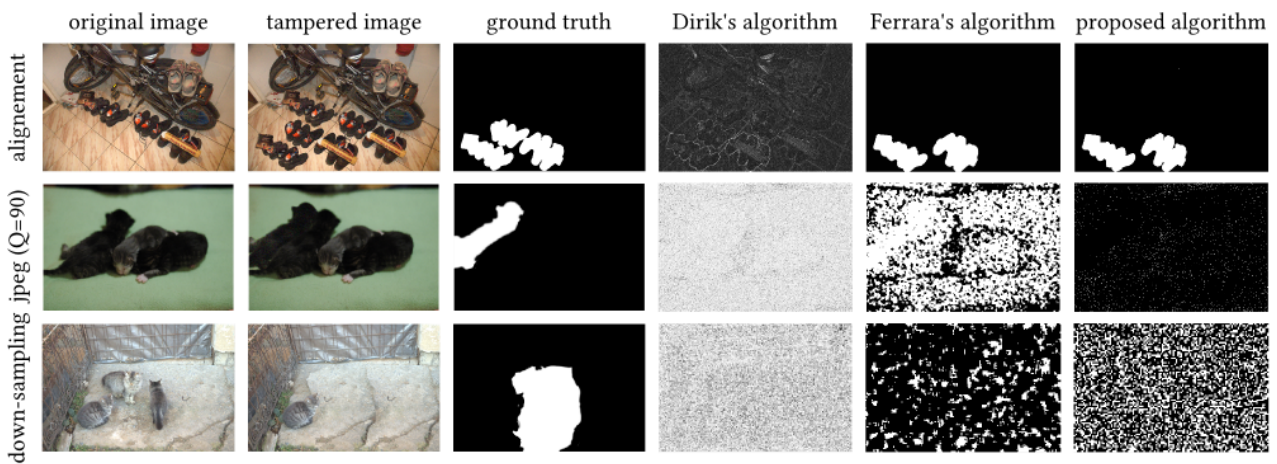
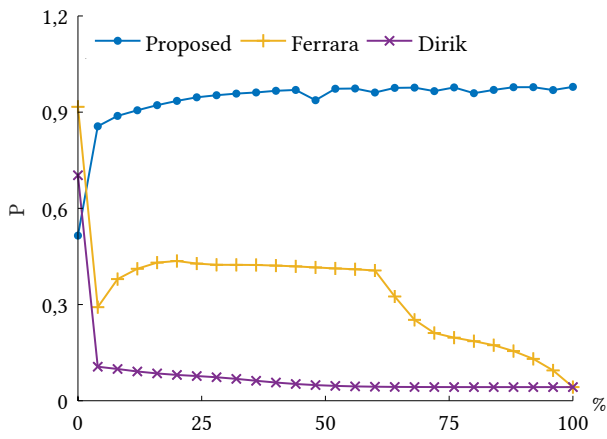
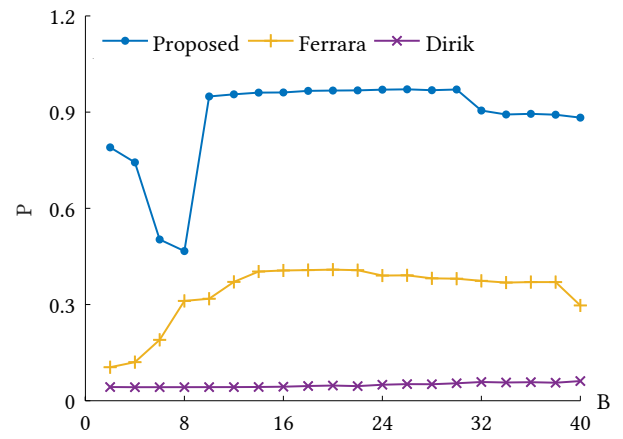


Figure 3.9: Examples of unsuccessful forgery localization by the proposed algorithm

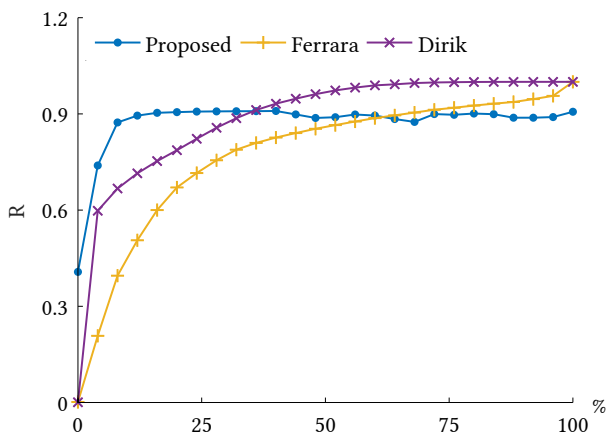
maps for two image sets associated with the Image Manipulation Dataset [41] and the Realistic Tampering Dataset [115] respectively. Each set consists of 20 random tampered images. From the localization maps and corresponding ground truths, we can derive TP , FN , FP , TN , and therefore compute P , R and F_1 . A comparative study on F_1 -Scores allows to evaluate the performance of the proposed



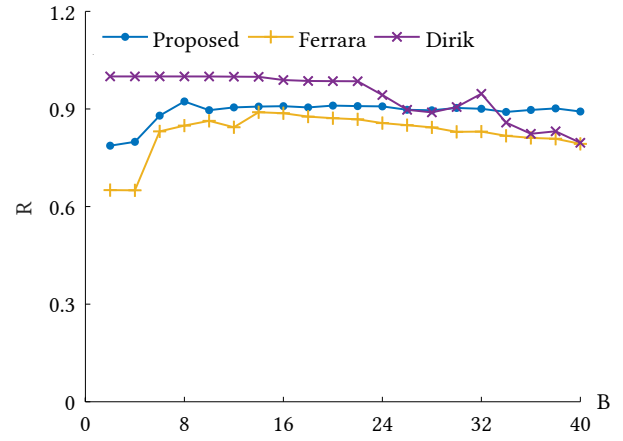
(a) Precision with respect to % of decision threshold



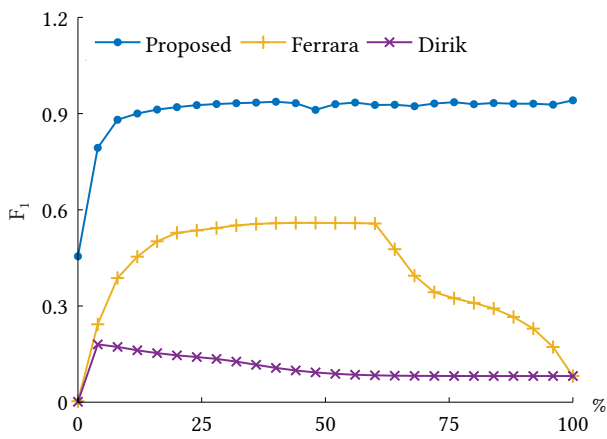
(b) Precision with respect to B



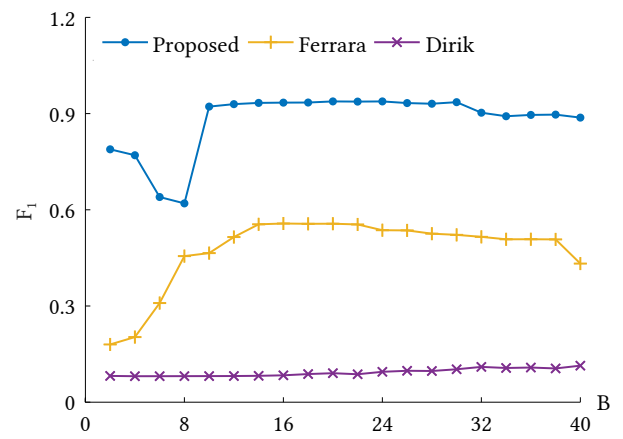
(c) Recall with respect to % of decision threshold



(d) Recall with respect to B



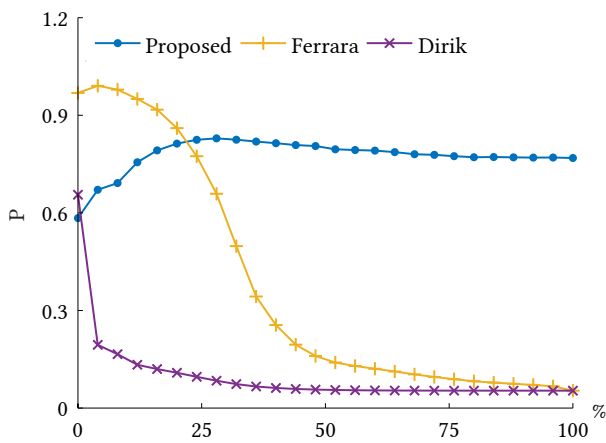
(e) F₁-Score with respect to % of decision threshold



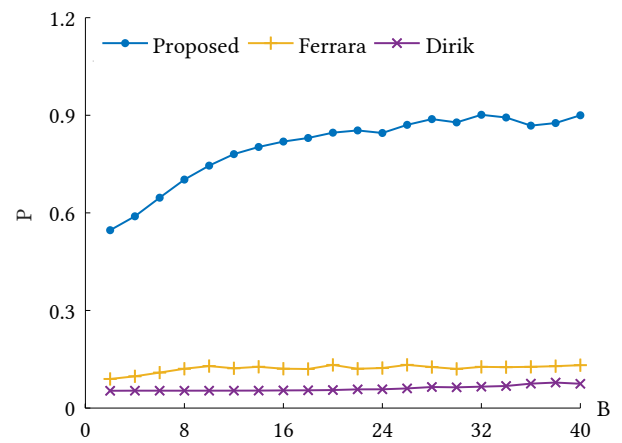
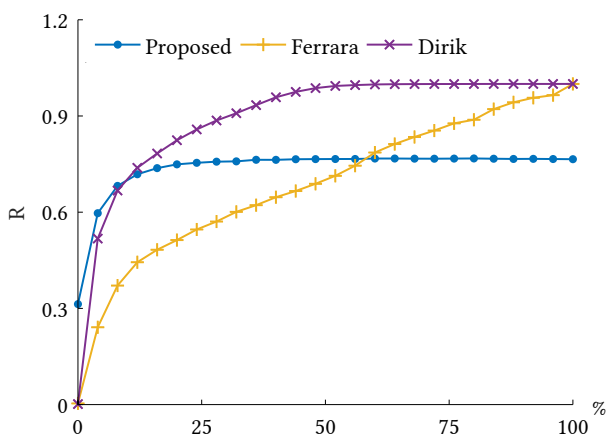
(f) F₁-Score with respect to B

Figure 3.10: Image Manipulation Dataset [41]

algorithm, as well as of the two benchmarks. Similarly to Section 3.5.2, two sensitivity studies are considered.



(a) Precision with respect to % of decision threshold

(b) Precision with respect to B 

(c) Recall with respect to % of decision threshold

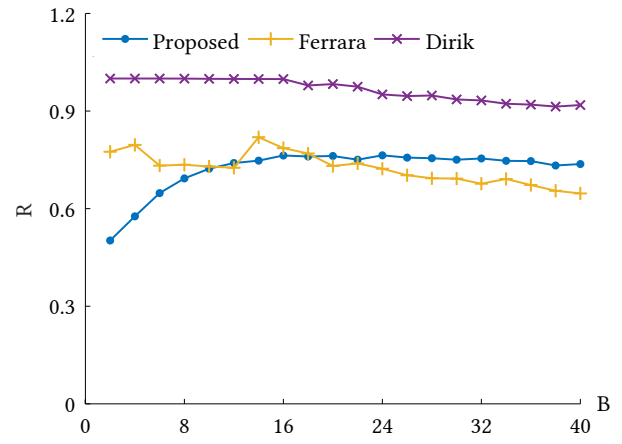
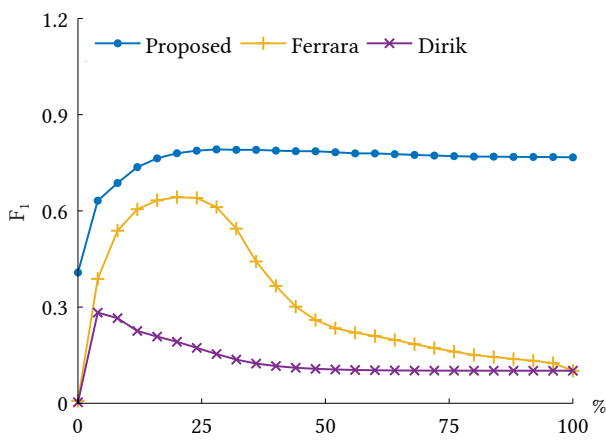
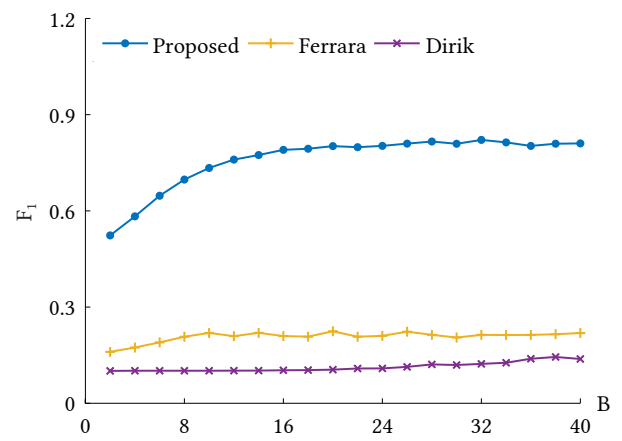
(d) Recall with respect to B (e) F_1 -Score with respect to % of decision threshold(f) F_1 -Score with respect to B

Figure 3.11: Realistic Tampering Dataset [115]

- *Sensitivity study 1:* B is fixed at 16, the values of τ and T vary. The threshold T in our algorithm takes value in the interval $[2, 52]$, while the threshold τ in the benchmark algorithms

takes value from the minimum to the maximum of variance maps. Note that, different variance maps of different images yield different ranges of values for τ . For comparison purpose, the thresholds value is further translated into percentage of associated intervals. Note that 0% and 100% correspond respectively to the minimum and maximum of each interval.

- *Sensitivity study 2*: The thresholds τ and T are fixed at 60% of ranges of values, and B varies from 2 to 40 with step 2.

Figures 3.10 and 3.11 show the results of these two studies when tested images are given from the Image Manipulation Dataset and Realistic Tampering Dataset respectively. In each figure, the sub-figures on the left and on the right display the evolution of P , R and F_1 with respect to % of decision threshold and to B . In all cases, we find that the F_1 -Score of the proposed algorithm is more stable at higher value than the benchmarks. This implies that our algorithm is more efficient and robust. Looking at the diagrams of P and R , we find that the high F_1 -Score of our algorithm mostly comes from the high value of P , because the values of R are more or less similar in the three algorithms. This results also mean that our algorithm allows a high probability that localized pixels are tampered, while the benchmark algorithms do not. This is totally consistent with the nature of non-threshold-based and threshold-based localization algorithms.

3.6 Conclusion

We develop in this chapter an improved algorithm for digital image authentication and forgery localization by jointly use the color filter array pattern identification, demosaicing algorithm estimation, and the local statistical analysis of demosaicing artifacts in spatial domain. A new feature statistic less sensitive to the edges problem is thus built to characterize demosaiced images. By modeling such feature statistics by a single normal mixture model for both tampered and untampered images, four well-known normality tests (i.e., Anderson-Darling test, one-sample Kolmogorov-Smirnov test, Jarque-Bera test and Lilliefors test) are employed to automatically authenticate digital images. Numerical experiments on the four well-known datasets shows that the performance of automatic authentication is relatively low, but can be much more improved thanks to human interpretation of supported graphic tools (i.e., Q-Q plot diagram, probability distribution curves, and localization map). Regarding the forgery localization, we propose a penalized EM algorithm to automatically distinguish between authentic and forged regions of a tampered image without any requirement on comparison thresholds as in most existing localization algorithm. Such a method is proved to be more effective and robust by numerical examples.

Chapter 4

Framework of JPEG DCT Coefficients Modeling and Forgery Localization

4.1 Introduction

Joint Photographic Experts Group (JPEG) is a lossy compression standard allowing a trade-off between memory size and visual quality for digital still images [213]. It is nowadays adopted by default in most digital cameras and Web services (e.g., social networks, photo galleries, etc.). However, this popularity also makes JPEG images to be the target of malicious attacks. This motivates us to develop in this chapter a reliable forensic tool to restore some trust to JPEG images. More especially, we are interested in the statistical modeling of *discrete cosine transform* (DCT) coefficients as the DCT is the basis of JPEG. Furthermore, our choice is to study these coefficients in their raw form rather than in modified forms as in some existing works (e.g., first and second significant digits of AC DCT coefficients [109], Benford-Fourier coefficients [166], etc.). To figure out better the problems to be dealt with, let us begin with an analysis of existing statistical models of AC DCT coefficients for JPEG images.

4.1.1 Prior Art

A great deal of effort has been put in building statistical models for *primary* unquantized AC DCT coefficients of *original* JPEG natural images. The pioneer is perhaps Pratt who fitted the statistical properties of primary unquantized AC DCT coefficients with a normal distribution (ND) four decades ago [177, chapter 10]. The choice of ND is conducted by the well-known central limit theorem. Afterwards, various empirical models have been proposed and verified by goodness-of-fit tests on standard images. For instance, Reininger and Gibson [188] relied on Kolmogorov-Smirnov (KS) tests to confirm that the Laplace distribution (LD) is more suitable than the ND for most images. Based on χ^2 tests, Müller [157] claimed that the generalized normal distribution (GND) approximates the statistics of unquantized AC DCT coefficients better than the LD. Using the same goodness-of-fit tests, Chang *et al.* recently reported in [30] that the generalized Gamma distribution (GGD) outperforms the LD and the GND. Cauchy distribution [56], α -stable distribution [24] and transparent composite model

[222] are other options. The main drawback of the above empirical models is the lack of mathematical foundation which makes their accuracy and robustness in question when applying to a wide range of images. To remedy this obstacle, Lam and Goodman analytically analyzed in [122] the statistical properties of unquantized AC DCT coefficients with respect to the variation of their block variance. They proved that these coefficients are reasonably modeled by a zero-mean ND when the block variance is fixed. However, as this variance is actually random for a JPEG natural image, a compound normal distribution¹ (CND) should be rather used. Such a distribution has been further extended in [158] by taking into account different flexible distributions for block variance. Yet, mathematical justifications for the block variance model were still missing till the appearance of the paper of Thai *et al.* [206]. They showed that the block variance of a JPEG natural image can be asymptotically approximated by a two-parameter Gamma distribution. This model is further applied in [207] and [189] to identify cameras model and estimate the quality factor of JPEG images respectively.

Statistical models of *quantized* AC DCT coefficients can be developed from the ones of primary unquantized coefficients by analyzing the JPEG compression chain. They can be classified according to the compression history of JPEG images. For single-compressed JPEG images, Qiao *et al.* characterized the quantized AC DCT coefficients by a LD in [178], and applied it to detect hidden information embedded by JSteg algorithm. A similar model was also adopted by Stamm and Liu in [198] to remove traces left by JPEG compression. Other models based on GND, GGD and CND can be found in [45, 206]. Compared to first quantized AC DCT coefficients, statistical models for the second ones are much scattered. In [220] and [204], those models are developed based on LD and CND respectively. Both of them are applied to estimate the first quantization steps in double-compressed JPEG images. For higher compression cycles, we found the works of Li *et al.* who developed statistical models for JPEG noises to retrieve the JPEG compression history in [132] and to distinguish between uncompressed and decompressed images in [131].

Besides *original* JPEG images, there exist *tampered* JPEG images whose original content has been modified by some localized forgery techniques such as splicing, cloning, cropping, etc. [114]. In such images, the *forged* and *unmodified* areas usually exhibit the properties of *single* and *double* JPEG compression respectively [171]. This is why mixture models are appropriate to statistically describe their AC DCT coefficients. In [18], Bianchi *et al.* used a two-components mixture model to approximate the histogram of *quantized* AC DCT coefficients of a tampered JPEG image. The mixing parameters was estimated by a simple Expectation-Maximization (EM) algorithm. The parameter estimation was further improved in [226] by exploiting the smoothness of the model likelihood function. Another improvement was proposed in [233] where quantization noise is eliminated before solving the mixture histogram model. More recently, Xue *et al.* developed in [221] a mixture model based on normalized gray level co-occurrence matrix to localize tampering in JPEG images. Such a model can take advantage of not only the double quantization effect inherent in tampered JPEG image but also the correlation among adjacent DCT blocks. If the aforementioned models are a mixture of the histograms of quantized AC DCT coefficients, Wang *et al.* developed in [217] an associated parametric mixture model whose components are derived from the LD of the primary unquantized coefficients.

¹Also called doubly stochastic model in [122].

4.1.2 Contributions and Organization

Through the above literature analysis, we find that statistical models developed for DCT coefficients of tampered JPEG images are still scattered and mostly limited to *double* quantization effect. Meanwhile, in many practical scenarios, images might be JPEG compressed several times (e.g., photos uploaded on blogs, online photo galleries, etc. [155]) before manipulated and saved again in JPEG format. Obviously, such images cannot be accurately characterized by aforementioned models. Therefore, our goal is to extend them to better deal with tampered JPEG images subject to *multiple* quantization effect. Furthermore, our choice is to study the DCT coefficients in their raw form directly rather than in modified forms as in some related works (e.g., first and second significant digits of DCT coefficients [109], Benford-Fourier coefficients [166], etc.). Consequently, we follow the JPEG compression process and the effects of manipulation operations on the DCT coefficients to develop a parametric statistical modeling framework for tampered JPEG images, and subsequently apply it to forgery localization. Although the modeling approach and forgery localization technique are relatively similar to some works in the literature (see e.g., [18, 217]), several significant improvements and contributions have been made.

1. We proposed a generalized modeling framework rather than a concrete model for DCT coefficients of tampered JPEG images. The framework allows to recursively derive any statistical model from a known distribution of primary unquantized DCT coefficients. By this way, the rich literature of primary unquantized DCT coefficients models is inherited to build a relevant statistical model that best fits with the histogram of quantized DCT coefficients of tampered JPEG images.
2. To describe the distribution of DCT coefficients of tampered JPEG images, most existing works relied on the family of mixture models. Nevertheless, no clear explanation has been given yet. For instance, how many components should be taken into account in the model is still an open issue. In this chapter, starting with a mixture of multiple components, we proved analytically that the histograms of quantized DCT coefficients of tampered JPEG images can be properly approximated by a *two-components* mixture model regardless of the number of forged areas in the images. The accuracy of such a simplified model is numerically assessed by the Kullback-Leibler divergence on various image databases.
3. About the forgery localization technique, we also relied on the Expectation-Maximization (EM) algorithm as in most related works. However, instead of estimating all the parameters of the model in the same time, we proposed to estimate the parameters of component's distribution, and the set of quantization steps and mixing coefficients separately. The former is estimated from *recovered* unquantized DCT coefficients of tampered JPEG images [68] by maximum likelihood (ML) estimation method, while the latter is obtained by applying the EM algorithm to the associated quantized DCT coefficients. This allows to speed up the forgery localization process.

Numerous numerical experiments under various configurations also confirms that our generalized modeling framework yields very promising performances when dealing with tampered JPEG images

subject to *multiple* quantization effect.

The remainder of this chapter is organized as follows. Section 4.2 focuses on developing a generalized statistical modeling framework for DCT coefficients of both the authentic and tampered JPEG images. The framework validation and accuracy assessment are performed in Section 4.3. In Section 4.4, we apply the developed framework in forgery localization for tampered JPEG images and evaluate its performance following the F_1 -score. Section 4.5 concludes the chapter. The models for primary unquantized DCT coefficients used in this chapter is also provided in Appendix B.

4.2 Recurrent Modeling Framework For JPEG DCT Coefficients

Let consider the tampering scenario where an authentic JPEG image which has undergone $i - 1$ compression times, $i \geq 2$, after some localized manipulation, is saved again without resizing in JPEG format with quality factor QF_i to generate a tampered JPEG image. The manipulation may be a splicing, a cloning or some post-processings (e.g., resize, rotation, etc.), that disrupt quantization structure of forged areas. So, after the last JPEG compression, the associated DCT coefficients likely present properties of a single compression with quality factor QF_i . Meanwhile, the DCT coefficients of the unmodified area exhibit properties of i compression times. Therefore, we first analyze the JPEG compression chain to find the statistical distributions of DCT coefficients at any compression cycle of authentic JPEG images. These distributions are next combined in a finite mixture paradigm to approximate the histogram of DCT coefficients of tampered JPEG images. A simplified version of this model is also proposed for practical uses.

4.2.1 Authentic JPEG Images

Generally, the main processing steps in a JPEG compression *chain* can be classified into two phases: *encoding* and *decoding* [213].

- In encoding phase, the uncompressed integer image I is first converted to a real floating-point representation W . Subdividing W into blocks 8×8 pixels and applying DCT operations to each of blocks separately, we obtain the unquantized DCT representation X of image with real coefficients. X is next quantized with a quantization table Q to yield the quantized DCT representation Y with integer coefficients. Finally, a lossless entropy encoding is applied on Y to generate the associated compressed image in JPEG format.
- The decoding phase is the reverse of the encoding phase. An entropy decoding is first applied on the compression image JPEG to return the quantized DCT representation \tilde{Y} with integer coefficients. \tilde{Y} is next dequantized using the same quantization table Q as in the encoding phase to yield the dequantized DCT representation \tilde{X} with integer coefficients. Applying inverse DCT on \tilde{X} , we get a new real floating-point representation \tilde{W} in spatial domain. The decoding phase ends with rounding and truncating operations on \tilde{W} to obtain a decompressed image \tilde{I} .

From the above description, we can define the i -th JPEG compression *cycle* as the process between two unquantized DCT representations $\mathbf{X}^{(i)}$ and $\mathbf{X}^{(i+1)}$, and schematize its processing diagram as in Figure 4.1a. Since this diagram is not convenient for statistical modeling, we propose to further sim-

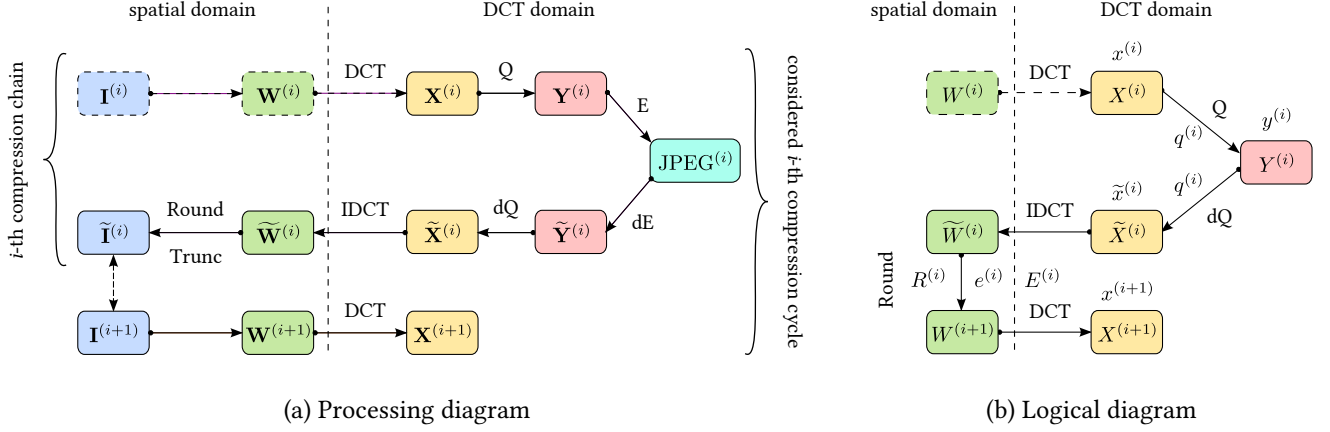


Figure 4.1: Main steps of the i -th JPEG compression cycle

plify this diagram by ignoring meaningless operations such as the entropy coding and decoding, the truncation effect, etc. It results in logical diagram as in Figure 4.1b. In this diagram, the capital letters $X^{(i)}$, $Y^{(i)}$ and $\tilde{X}^{(i)}$ stand for the random unquantized, quantized and dequantized DCT coefficients at a given frequency, while the smaller letters $x^{(i)}$, $y^{(i)}$ and $\tilde{x}^{(i)}$ are the associated realizations. The random rounding error $R^{(i)}$ in spatial domain correspond to the random error $E^{(i)}$ in DCT domain with realization $e^{(i)}$. Finally, the notation $q^{(i)}$ denotes the quantization step associated with the compression quality factor $Q^{F^{(i)}}$. Note that the considered frequency is ignored from the above notations to further simplify. Based on the logical diagram in Figure 4.1b, we would like to statistically characterize the DCT coefficients $Y^{(i)}$, $\tilde{X}^{(i)}$ and $X^{(i+1)}$ from the distribution of $X^{(i)}$.

Passing through the quantization operation with step $q^{(i)}$, the continuous random variable $X^{(i)} \in \mathbb{R}$ becomes the integer random variable $Y^{(i)} \in \mathbb{Z}$, such that

$$Y^{(i)} = \left\lfloor \frac{X^{(i)}}{q^{(i)}} \right\rfloor, \quad (4.1)$$

where $\lfloor \cdot \rfloor$ denotes the nearest integer rounding. This leads to the following equivalence

$$\{Y^{(i)} = y^{(i)}\} \Leftrightarrow \left\{X^{(i)} \in \left[q^{(i)}y^{(i)} - \frac{q^{(i)}}{2}, q^{(i)}y^{(i)} + \frac{q^{(i)}}{2} \right) \right\}. \quad (4.2)$$

We can therefore compute the *probability mass function* (pmf) of $Y^{(i)}$ at $y^{(i)}$ as

$$p_{Y^{(i)}}(y^{(i)}; \boldsymbol{\theta}, \mathbf{q}^{(i)}) = F_{X^{(i)}}\left(q^{(i)}y^{(i)} + \frac{q^{(i)}}{2}; \boldsymbol{\theta}, \mathbf{q}^{(i-1)}\right) - F_{X^{(i)}}\left(q^{(i)}y^{(i)} - \frac{q^{(i)}}{2}; \boldsymbol{\theta}, \mathbf{q}^{(i-1)}\right), \quad (4.3)$$

where $F_{X^{(i)}}(x^{(i)}; \boldsymbol{\theta}, \mathbf{q}^{(i-1)})$ denotes the *cumulative distribution function* (cdf) of the unquantized DCT coefficient $X^{(i)}$ at $x^{(i)} \in \mathbb{R}$ with the model parameter $\boldsymbol{\theta}$ and the set of quantization steps $\mathbf{q}^{(i-1)} =$

$(q^{(1)}, \dots, q^{(i-1)})$ of $i - 1$ previous JPEG compression cycles.

Dequantizing $Y^{(i)}$ with the same step $q^{(i)}$, we obtain the integer random variable $\tilde{X}^{(i)}$ such that

$$\tilde{X}^{(i)} = q^{(i)} \cdot Y^{(i)}, \quad (4.4)$$

where $\tilde{X}^{(i)} \in \mathbb{Z}_{q^{(i)}} \triangleq q^{(i)} \times \mathbb{Z} = \{0, \pm q^{(i)}, \pm 2q^{(i)}, \pm 3q^{(i)}, \dots\}$. Using (4.3), the pmf of $\tilde{X}^{(i)}$ at $\tilde{x}^{(i)}$ is given by

$$p_{\tilde{X}^{(i)}}(\tilde{x}^{(i)}; \boldsymbol{\theta}, \mathbf{q}^{(i)}) = F_{X^{(i)}}\left(\tilde{x}^{(i)} + \frac{q^{(i)}}{2}; \boldsymbol{\theta}, \mathbf{q}^{(i-1)}\right) - F_{X^{(i)}}\left(\tilde{x}^{(i)} - \frac{q^{(i)}}{2}; \boldsymbol{\theta}, \mathbf{q}^{(i-1)}\right), \quad (4.5)$$

where $\tilde{x}^{(i)} \in \mathbb{Z}_{q^{(i)}}$.

Now, passing through the successive {IDCT, Rounding, DCT} operations, $\tilde{X}^{(i)} \in \mathbb{Z}_{q^{(i)}}$ becomes $X^{(i+1)} \in \mathbb{R}$. We can therefore express the continuous random variable $X^{(i+1)}$ as

$$X^{(i+1)} = \text{DCT}\left(\left\lfloor \text{IDCT}\left(\tilde{X}^{(i)}\right)\right\rfloor\right) = \text{DCT}\left(\text{IDCT}\left(\tilde{X}^{(i)}\right) + R^{(i)}\right) = \tilde{X}^{(i)} + E^{(i)}, \quad (4.6)$$

where $R^{(i)}$ denotes the error introduced by rounding the output values to 8 bit integers in the spatial domain, and $E^{(i)} = \text{DCT}\left(R^{(i)}\right) \in \mathbb{R}$ is the associated rounding error in DCT domain. It is widely admitted in the literature (see e.g., [143]) that $E^{(i)}$ follows a zero-mean normal distribution (ND) with variance $\frac{1}{12}$ (i.e., $E^{(i)} \sim \mathcal{N}\left(0, \frac{1}{12}\right)$). So, the cdf and pdf of $E^{(i)}$ at $e^{(i)}$ are respectively by

$$F_{E^{(i)}}\left(e^{(i)}\right) = \Phi\left(e^{(i)}\sqrt{12}\right) \quad \text{and} \quad f_{E^{(i)}}\left(e^{(i)}\right) = \sqrt{12}\varphi\left(e^{(i)}\sqrt{12}\right), \quad (4.7)$$

where $\Phi(z)$ and $\varphi(z)$ stand for the cdf and pdf of standard ND such that

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-\frac{u^2}{2}} du \quad \text{and} \quad \varphi(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}. \quad (4.8)$$

We can now compute the cdf of $X^{(i+1)}$ at $x^{(i+1)}$ from (4.6) by

$$F_{X^{(i+1)}}\left(x^{(i+1)}; \boldsymbol{\theta}, \mathbf{q}^{(i)}\right) = \sum_{\tilde{x}^{(i)} \in \mathbb{Z}_{q^{(i)}}} \Phi\left(\left(x^{(i+1)} - \tilde{x}^{(i)}\right)\sqrt{12}\right) \times \left(F_{X^{(i)}}\left(\tilde{x}^{(i)} + \frac{q^{(i)}}{2}; \boldsymbol{\theta}, \mathbf{q}^{(i-1)}\right) - F_{X^{(i)}}\left(\tilde{x}^{(i)} - \frac{q^{(i)}}{2}; \boldsymbol{\theta}, \mathbf{q}^{(i-1)}\right)\right), \quad (4.9)$$

where $\Phi(\cdot)$ is given by (4.8). Differentiating $F_{X^{(i+1)}}\left(x^{(i+1)}; \boldsymbol{\theta}, \mathbf{q}^{(i)}\right)$ with respect to $x^{(i+1)}$, we obtain the pdf of $X^{(i+1)}$ at $x^{(i+1)}$ as

$$f_{X^{(i+1)}}\left(x^{(i+1)}; \boldsymbol{\theta}, \mathbf{q}^{(i)}\right) = \sqrt{12} \sum_{\tilde{x}^{(i)} \in \mathbb{Z}_{q^{(i)}}} \varphi\left(\left(x^{(i+1)} - \tilde{x}^{(i)}\right)\sqrt{12}\right) \times \left(F_{X^{(i)}}\left(\tilde{x}^{(i)} + \frac{q^{(i)}}{2}; \boldsymbol{\theta}, \mathbf{q}^{(i-1)}\right) - F_{X^{(i)}}\left(\tilde{x}^{(i)} - \frac{q^{(i)}}{2}; \boldsymbol{\theta}, \mathbf{q}^{(i-1)}\right)\right), \quad (4.10)$$

with $\varphi(z)$ given from (4.8).

Equations (4.3), (4.5), (4.9) and (4.10) constitutes the recurrent framework to derive statistical models for all kinds of DCT coefficients of an authentic JPEG image. In other words, we can recursively obtain the statistical law of any coefficient if the cdf $F_{X^{(1)}}(x^{(1)}; \boldsymbol{\theta})$ (or the pdf $f_{X^{(1)}}(x^{(1)}; \boldsymbol{\theta})$) of primary unquantized DCT coefficients $X^{(1)}$ is known. Also, some well-known statistical models of $X^{(1)}$ can be found in Appendix B.

4.2.2 Tampered JPEG Images

Consider now a tampered JPEG image with K forged areas, the pmf of its quantized DCT coefficients Y at y is naturally described by the mixture model

$$p_Y(y; \boldsymbol{\Theta}_K, \boldsymbol{\Lambda}_K, \mathbf{q}^{(i)}) = \lambda_0 \cdot p_{Y^{(i)}}(y; \boldsymbol{\theta}_0, \mathbf{q}^{(i)}) + \sum_{k=1}^K \lambda_k \cdot p_{Y^{(1)}}(y; \boldsymbol{\theta}_k, q^{(i)}), \quad (4.11)$$

where $\boldsymbol{\Theta}_K = (\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K)$ stands for the set of parameters of unmodified and K forged areas respectively, $\boldsymbol{\Lambda}_K = (\lambda_0, \lambda_1, \dots, \lambda_K)$, with $\lambda_k > 0$ and $\sum_{k=0}^K \lambda_k = 1$, denotes the set of corresponding mixing coefficients. The pmf $p_{Y^{(i)}}(y; \boldsymbol{\theta}_0, \mathbf{q}^{(i)})$ implies that the *unmodified* area characterized by the parameter $\boldsymbol{\theta}_0$ has been undergone i JPEG compression times with quantization steps $\mathbf{q}^{(i)} = (q^{(1)}, \dots, q^{(i)})$. Meanwhile, $p_{Y^{(1)}}(x; \boldsymbol{\theta}_k, q^{(i)})$, $k = 1, \dots, K$, means that K *forged* areas characterized by the parameters λ_k have been likely compressed once with quantization step $q^{(i)}$ regardless of the compression history of their source [138]. Both $p_{Y^{(i)}}(y; \boldsymbol{\theta}_0, \mathbf{q}^{(i)})$ and $p_{Y^{(1)}}(y; \boldsymbol{\theta}_k, q^{(i)})$ can be obtained by the recurrent modeling framework in Subsection 4.2.1. Despite its natural interpretation, the model (4.11) faces with two main drawbacks. Firstly, the number of forged areas K is unknown in practice. Although some unsupervised learning techniques of finite mixture models (see e.g., [66] and references therein) can help, they are usually inefficient in the case of DCT coefficients because of very similar distributions of components. Therefore, how to identify K is still an open issue. Secondly, even if K is already known, a large number of model parameters (i.e., $\boldsymbol{\Theta}_K$, $\boldsymbol{\Lambda}_K$ and $\mathbf{q}^{(i)}$) are obviously not easy to deal with. To overcome these obstacles, a simplification of model (4.11) is proposed hereinafter.

To start with, we characterize the statistical properties of primary unquantized DCT coefficients associated with the considered tampered JPEG image. Similarly to (4.11), their pdf is given by

$$f_{X^{(1)}}(x; \boldsymbol{\Theta}_K, \boldsymbol{\Lambda}_K) = \lambda_0 \cdot f_{X^{(1)}}(x; \boldsymbol{\theta}_0) + \sum_{k=1}^K \lambda_k \cdot f_{X^{(1)}}(x; \boldsymbol{\theta}_k), \quad (4.12)$$

where $f_{X^{(1)}}(x; \boldsymbol{\theta}_0)$ and $f_{X^{(1)}}(x; \boldsymbol{\theta}_k)$, $k = 0, \dots, K$, represent the same pdf form of primary unquantized AC DCT coefficients for the unmodified area and K forged areas respectively. As $f_{X^{(1)}}(x; \boldsymbol{\theta}_0)$ and $f_{X^{(1)}}(x; \boldsymbol{\theta}_k)$ are unknown in practice, we should find another way to express $f_{X^{(1)}}(x; \boldsymbol{\Theta}_K, \boldsymbol{\Lambda}_K)$. Inspired by the work of Fridrich *et al.* [68], we follow the recovery process of primary unquantized AC DCT coefficients in Figure 4.2. Firstly, we decode the tampered JPEG image, then crop r rows and c

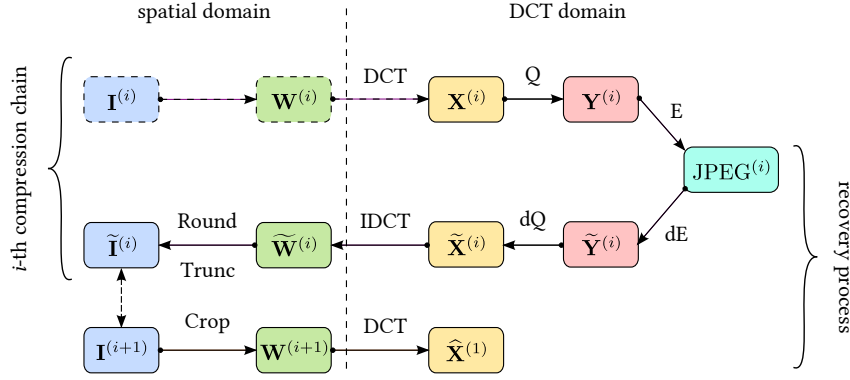


Figure 4.2: Recovery process of primary unquantized AC DCT coefficients

columns, $1 \leq r, c \leq 7$, in the spatial domain to break the quantized structure, and finally convert the cropped image again in DCT domain to recover unquantized DCT coefficients. From such recovered coefficients of the considered tampered JPEG image, we can get an approximation of $f_{X^{(1)}}(x; \Theta_K, \Lambda_K)$. We also note that similar results are obtained when the cropping in the above method is replaced by a slight amount of rotation or resizing. Besides, the wavelet denoising method proposed in [88] could be used for the same purpose. Although no analytical proof is given for this method, numerous numerical experiments show that its performance is well adequate [112]. In other words, $f_{X^{(1)}}(x; \Theta_K, \Lambda_K)$ can be properly approximated by a single parametric distribution as

$$f_{X^{(1)}}(x; \Theta_K, \Lambda_K) \simeq f_{X^{(1)}}(x; \hat{\theta}), \quad (4.13)$$

where $\hat{\theta}$ is estimated from *recovered* unquantized DCT coefficients by a classical estimation method (e.g., the ML method described in Appendix B). Intuitively, we can explain in part this phenomenon by the similarity in the pdf forms of unquantized AC DCT coefficients and by small portion of forged areas (see also Figure 4.5). Besides, the image cropping has almost no effect on the portions $\lambda_0, \dots, \lambda_K$ of unmodified and forged areas. Of course, the approximation quality depends closely on the flexibility of the considered probability law. Mimicking (4.12), we rewrite (4.13) in the form

$$f_{X^{(1)}}(x; \Theta_K, \Lambda_K) \simeq \lambda_0 \cdot f_{X^{(1)}}(x; \hat{\theta}) + \sum_{k=1}^K \lambda_k \cdot f_{X^{(1)}}(x; \hat{\theta}) = \lambda_0 \cdot f_{X^{(1)}}(x; \hat{\theta}) + (1 - \lambda_0) \cdot f_{X^{(1)}}(x; \hat{\theta}), \quad (4.14)$$

as $\sum_{k=0}^K \lambda_k = 1$. Therefore, an approximation of (4.11) is given by

$$p_Y(y; \Theta_K, \Lambda_K, \mathbf{q}^{(i)}) \simeq \lambda_0 \cdot p_{Y^{(i)}}(y; \hat{\theta}, \mathbf{q}^{(i)}) + (1 - \lambda_0) \cdot p_{Y^{(1)}}(y; \hat{\theta}, \mathbf{q}^{(i)}) \triangleq p_Y(y; \hat{\theta}, \lambda_0, \mathbf{q}^{(i)}). \quad (4.15)$$

We can remark that the simplified model (4.15) allows to bypass difficulties inherent in the models (4.11): *the model is now invariant irrespective of the number of source images causing forged areas K , and the parameters is now limited and can be estimated easily*. The price paid for this simplicity is that we can no longer rely on the component's parameters (i.e., $\theta_0, \dots, \theta_K$) to distinguish between

unmodified and forged areas, and the difference in their quantization effects is the only measure for their distinction (see also Subsection 4.4.2).

4.3 Framework Validation and Accuracy Assessment

The goals of this section are two-fold: (i) numerical validation of the modeling framework when primary unquantized AC DCT coefficients are respectively modeled by the Laplace distribution (LD), the generalized normal distribution (GND), the generalized gamma distribution (GGD), and the compound normal distribution² (CND) (see also Appendix B for more details), and (ii) accuracy assessment of the simplified model (4.15) under different configurations of tampered JPEG images. In all cases, the evaluation is performed via the Kullback-Leibler (K-L) divergence.

4.3.1 Dataset Building

For numerical experiments, we built a tampered image dataset from 6 well-known uncompressed color image databases with different image sizes: McMaster [230] (500×500), Standard and CSIQ [123] (512×512), Kodak³ (768×512), McGill [162] (768×576) and Realistic [115] (1920×1080). For each database, images are randomly JPEG compressed one or multiple times (by e.g., using `imwrite` function in Matlab). These images are next combined together following the portions $\lambda_0, \dots, \lambda_K$ such that $\sum_{k=0}^K \lambda_k = 1$, and saved again in JPEG format to generate tampered JPEG images. The combination technique is either cloning or splicing so that the DCT grid of forged areas is misaligned with the one of the host image. Figures 4.3a and 4.3b illustrate respectively two typical examples of tampered JPEG images when the quality factor of the last JPEG compression is $QF_3 = 70$ and $QF_3 = 90$ respectively. In

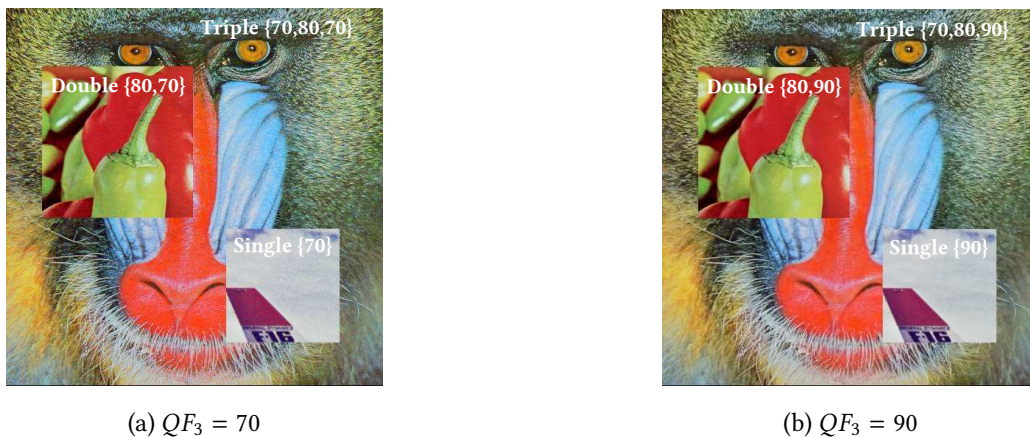


Figure 4.3: Two typical examples of tampered JPEG images

both cases, the same source images are given from the Standard database: a double JPEG image (given by successively compressing Baboon.png with quality factors 70 and 80), a single JPEG image (given

²Also called doubly stochastic model in [122].

³<http://r0k.us/graphics/kodak/>

by compressing Peppers.png with quality factor 80), and an uncompressed image (Airplane.png). After pasting a square area of Peppers and Airplane with ratios $\lambda_1 = 0.16$ and $\lambda_2 = 0.09$ to Baboon, and next compressing with quality factor 70 and 90 respectively, we obtain the tampered images. These two tampered images will be used hereafter for other numerical illustrations.

The compression number i of unmodified area, the associated quality factors (QF_1, \dots, QF_i) , the number of forged areas K and their portions $\lambda_1, \dots, \lambda_K$ define a configuration of tampered JPEG images. In our numerical experiments, each concrete configuration is experimented on 720 images (i.e., 120 images for each of 6 aforementioned databases). Since 36 configurations are studied in Subsections 4.3.2 and 4.4.2, we built in total 25920 tampered JPEG images. Some of them are displayed in Figure 4.4.



Figure 4.4: Examples of tampered JPEG images extracted from the built dataset (from left to right: McMaster, Stand, CSIQ, Kodak, McGill, Realistic)

4.3.2 Framework Validation

The approach to validate the modeling framework is to compare the empirical pmf of quantized AC DCT coefficients $\hat{p}_Y(y)$ with their true and simplified models $p_Y(y; \Theta_K, \Lambda_K, \mathbf{q}^{(i)})$ and $p_Y(y; \hat{\theta}, \lambda_0, \mathbf{q}^{(i)})$. The parameters Θ_K and $\hat{\theta}$ in these models are respectively estimated from uncompressed source images and tampered image using the ML method described in Appendix B. Meanwhile, for simplification, Λ_K and $\mathbf{q}^{(i)}$ are merely assumed known (an EM algorithm will be introduced in Section 4.4 to find their estimates).

Let continue with the two tampered JPEG images in Figures 4.3a and 4.3b, we would like to study the pmf of their quantized DCT coefficients at the 2nd AC frequency⁴ under the assumption that the associated primary unquantized DCT coefficients follow a GGD such that $X^{(1)} \sim \text{GGD}(\alpha, \beta, \gamma)$. To this end, we specify the model parameters as in Table 4.1, and sketch in Figures 4.5a and 4.5b the pdf of primary unquantized coefficients $X^{(1)}$ when $QF_3 = 70$ and $QF_3 = 90$ respectively. In each figure, the pdf associated with source images $f_{X^{(1)}}(x; \theta_k)$, $k = 0, 1, 2$, are represented on top, while the ones related to tampered image $f_{X^{(1)}}(x; \Theta_2, \Lambda_2)$ and $f_{X^{(1)}}(x; \hat{\theta})$ are represented at the bottom.

Table 4.1 shows clearly that for whatever QF_3 , the values of $\hat{\theta}$ estimated from tampered JPEG images are closed. Moreover, as displayed in Figures 4.5a and 4.5b, the mixture $f_{X^{(1)}}(x; \Theta_2, \Lambda_2)$ can be well approximated by a single pdf $f_{X^{(1)}}(x; \hat{\theta})$. This confirms our conjecture in (4.13). Now, we sketched in Figures 4.6a and 4.6b the pmf of the associated quantized coefficients. In each figure, the true pmf $p_Y(y; \Theta_2, \Lambda_2, \mathbf{q}^{(3)})$ on top, and the simplified pmf $p_Y(y; \hat{\theta}, \lambda_0, \mathbf{q}^{(3)})$ are displayed at the bottom. Both of them well match the empirical pmf $\hat{p}_Y(y)$. Such a similarity justifies the correctness of the developed

⁴The position (2, 1) of 8×8 DCT blocks

parameters	uncompressed source image			tampered JPEG image	
	Baboon	Peppers	Airplane	$QF_3 = 70$	$QF_3 = 90$
$\hat{\alpha}$	1.0709	8.8860	4.5465	1.5142	1.6037
$\hat{\beta}$	44.4930	0.0014	0.1002	21.8281	19.2895
$\hat{\gamma}$	0.9230	0.2296	0.2834	0.6886	0.6657
$q^{(3)}$	—	—	—	7	2
others	$\lambda_0 = 0.75, \lambda_1 = 0.16, \lambda_2 = 0.09, q^{(1)} = 7, q^{(2)} = 5$				

Table 4.1: $\Theta_2 = (\theta_0, \theta_1, \theta_2)$, $\hat{\theta} = (\hat{\alpha}, \hat{\beta}, \hat{\gamma})$, $\Lambda_2 = (\lambda_0, \lambda_1, \lambda_2)$ and $\mathbf{q}^{(3)} = (q^{(1)}, q^{(2)}, q^{(3)})$

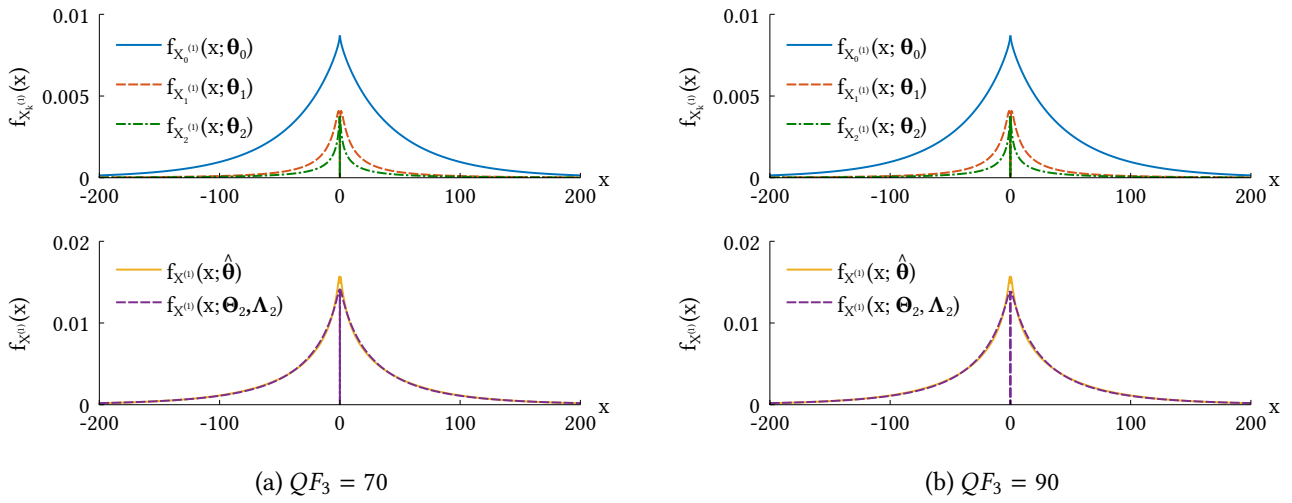


Figure 4.5: Shapes of $f_{X^{(1)}}(x; \Theta_2, \Lambda_2)$ and $f_{X^{(1)}}(x; \hat{\theta})$

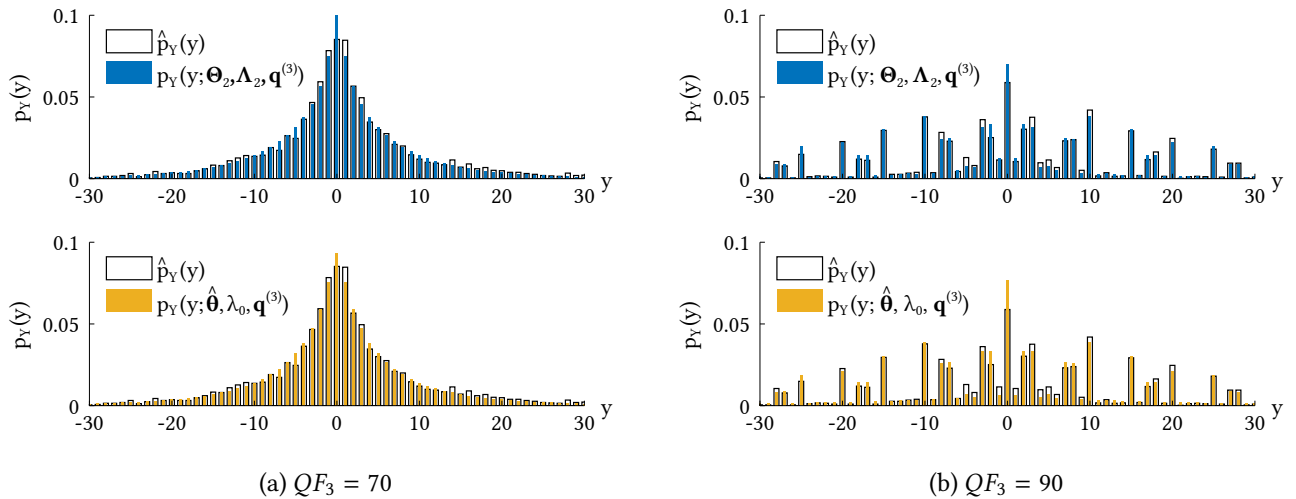


Figure 4.6: Shapes of $\hat{p}_Y(y)$, $p_Y(y; \Theta_2, \Lambda_2, \mathbf{q}^{(3)})$ and $p_Y(y; \hat{\theta}, \lambda_0, \mathbf{q}^{(3)})$

modeling framework for tampered JPEG images. We note that various numerical experiments on other JPEG tampered images give the same conclusions.

The developed framework requires a suitable pdf for primary unquantized DCT coefficients. We would like to find out which one among the well-known LD, GND, GGD and CGD allows the most accurate model for tampered JPEG images. We use therefore the K-L divergence to evaluate the differences in the empirical pmf $\hat{p}_Y(y)$ and its models $p_Y(y; \Theta)$

$$D_{KL}(\hat{p}_Y(y) \parallel p_Y(y; \Theta)) = \sum_{y \in \mathbb{Z}} \hat{p}_Y(y) \cdot \ln \left(\frac{\hat{p}_Y(y)}{p_Y(y; \Theta)} \right), \quad (4.16)$$

where $p_Y(y; \Theta)$ is given from either the true model (4.11) or the simplified model (4.15). This choice is motivated by a good compromise between the fitness of the main portion and the fitness of the tail part returned by the K-L divergence [222]. More precisely, the evaluation is performed on 720 tampered JPEG images with 2 forged areas: one from an uncompressed image with portion $\lambda_1 = 0.09$ and another from a single JPEG image with quality factor 80 and portion $\lambda_2 = 0.16$. The host image

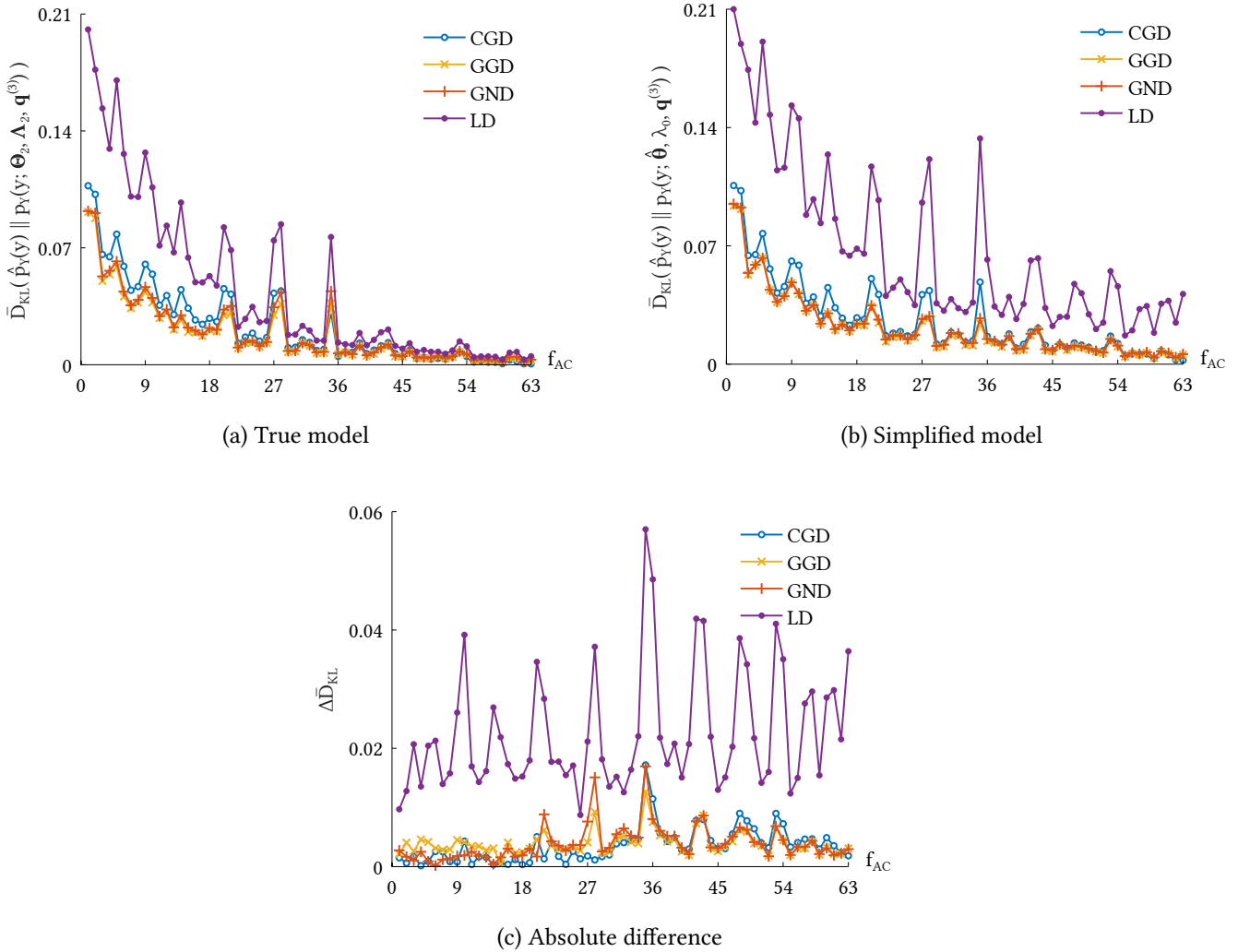


Figure 4.7: Average K-L divergence for different pdf of primary unquantized AC DCT coefficients is double compressed with quality factors 70 and 80, and the JPEG compression after manipulation

is with quality factor $QF_3 = 90$. By varying the AC frequency of quantized AC DCT coefficients from 1 to 63 in zig-zag order and considering respectively LD, GND, GGD and CND as model for primary unquantized coefficients, we sketch in Figures 4.7a, 4.7b and 4.7c the average K-L divergences $\bar{D}_{KL}(\hat{p}_Y(y) \parallel p_Y(y; \Theta_2, \Lambda_2, \mathbf{q}^{(3)}))$, $\bar{D}_{KL}(\hat{p}_Y(y) \parallel p_Y(y; \hat{\theta}, \lambda_0, \mathbf{q}^{(3)}))$, and their absolute difference $\Delta\bar{D}_{KL}$ respectively. In these figures, a lower curve correspond to a higher accuracy. As such, the GGD-based models and LD-based models are respectively the best and the worse. The models based on GND and CND are also good alternatives because their average K-L divergences are close to the one of the GGD-based model. In all models, the accuracy is better at high AC frequency. However, as we shall see in Section 4.4, this high accuracy is not really useful for forgery localization, because the DCT coefficients of both unmodified and forged areas are quantized mostly at 0. We also find in Figure 4.7c that the simplified model (4.15) is well comparable to the true model (4.11) in the case of GND, GGD and CND, because the associated absolute differences are very close to 0. But, it loses the accuracy when the LD is used as primary distribution. All these results confirm that both the true and simplified models characterize correctly a tampered JPEG image if the distribution of primary AC DCT coefficients is properly chosen.

Now, we study the impact of JPEG compression times on the modeling framework. Three configurations of tampered JPEG images associated with the (i) *double* compression ($QF_1 = 75, QF_2 = 80$), (ii) *triple* compression ($QF_1 = 75, QF_2 = 80, QF_3 = 85$), and (iii) *quadruple* compression ($QF_1 = 75, QF_2 = 80, QF_3 = 85, QF_4 = 90$) of unmodified area are thus considered. We experiment each configuration on 720 tampered JPEG images with unmodified portion $\lambda_0 = 0.75$ and 3 forged areas (i.e., $K = 3$). To simplify, all forged areas are of the same size (i.e., $\lambda_1 = \lambda_2 = \lambda_3 \simeq 0.0833$) and given from single JPEG images with quality factor 75. Moreover, we only compute the K-L divergence for the case of GND as it gives the best compromise between the model accuracy and the computation time. As a result, we obtain in Figure 4.8 the curves of average K-L divergence with respect to 63 AC frequencies in zig-zag order. Their meanings are just similar to the ones in Figure 4.7. We can remark that both the true and simplified models always give a higher K-L divergence when the compression number i increases. This is mainly due to the truncated error is ignored from the modeling of JPEG compression cycle. Still, these differences are very small and do not much influence the modeling accuracy.

4.3.3 Accuracy Assessment

To see how good the simplified model (4.15) approximates the true model (4.11), we consider the GND as the model of primary unquantized AC DCT coefficients, and evaluate the mean μ_{KL}^{63} and variance v_{KL}^{63} over 63 AC frequencies of the K-L divergences

$$\mu_{KL}^{63} = E \left[D_{KL} \left(p_Y \left(y; \Theta_K, \Lambda_K, \mathbf{q}^{(i)} \right) \parallel p_Y \left(y; \hat{\theta}, \lambda_0, \mathbf{q}^{(i)} \right) \right) \right], \quad (4.17)$$

and

$$v_{KL}^{63} = \text{var} \left[D_{KL} \left(p_Y \left(y; \Theta_K, \Lambda_K, \mathbf{q}^{(i)} \right) \parallel p_Y \left(y; \hat{\theta}, \lambda_0, \mathbf{q}^{(i)} \right) \right) \right], \quad (4.18)$$

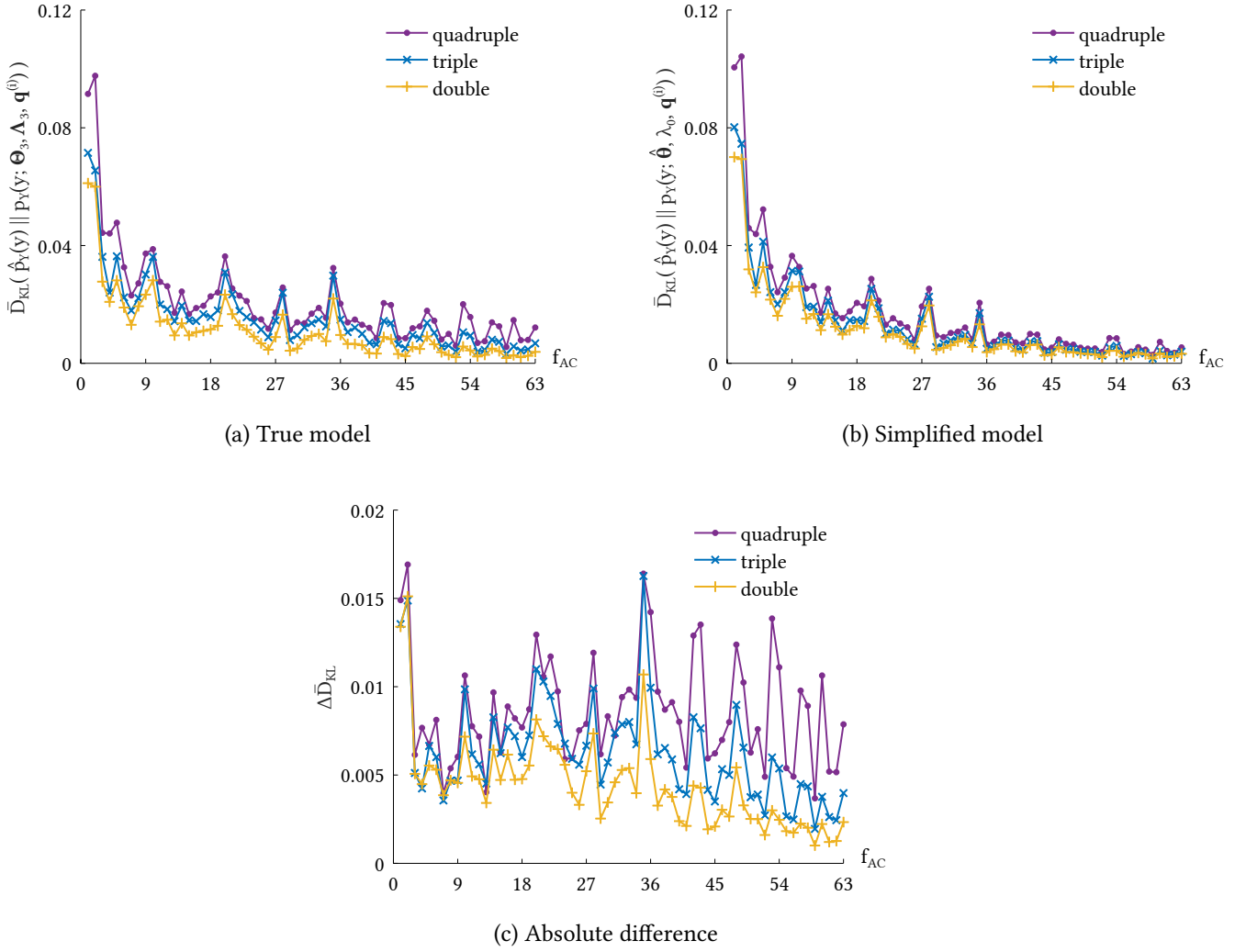


Figure 4.8: Average K-L divergence for different JPEG compression numbers

where

$$D_{KL} \left(p_Y \left(y; \Theta_K, \Lambda_K, \mathbf{q}^{(i)} \right) \| p_Y \left(y; \hat{\Theta}, \lambda_0, \mathbf{q}^{(i)} \right) \right) = \sum_{y \in \mathbb{Z}} p_Y \left(y; \Theta_K, \Lambda_K, \mathbf{q}^{(i)} \right) \cdot \ln \left(\frac{p_Y \left(y; \Theta_K, \Lambda_K, \mathbf{q}^{(i)} \right)}{p_Y \left(y; \hat{\Theta}, \lambda_0, \mathbf{q}^{(i)} \right)} \right) \quad (4.19)$$

under various configurations of tampered JPEG images. In all configurations, the unmodified area of tampered images is triple-compressed with (QF_1, QF_2, QF_3) , while the forged areas are randomly given from uncompressed, single-compressed or multiple-compressed JPEG images with quality factors taken from the set $[75, 80, \dots, 95]$. To further simplify, all forged areas inside a tampered image are set at the same size.

To understand the impact of λ_0 , we fix $K = 2$ and $(QF_1, QF_2, QF_3) = (70, 80, 90)$ (i.e., $i = 3$), vary λ_0 from 0.7 to 0.9 with step 0.05, and sketch μ_{KL}^{63} and ν_{KL}^{63} in Figure 4.9a. A higher λ_0 implies a smaller portion for all forged areas. Since a single pdf $f_{X^{(i)}}(x; \hat{\Theta})$ approximates the mixture pdf $f_{X^{(i)}}(x; \Theta_K, \Lambda_K)$

better in this case, it is not surprising that $p_Y(y; \hat{\theta}, \lambda_0, \mathbf{q}^{(i)})$ moves closer to $p_Y(y; \Theta_K, \Lambda_K, \mathbf{q}^{(i)})$. This explains why both the μ_{KL}^{63} and v_{KL}^{63} decrease when λ_0 increases. In other words, the simplified model (4.15) is more accurate for higher portion of unmodified area, that is actually logical.

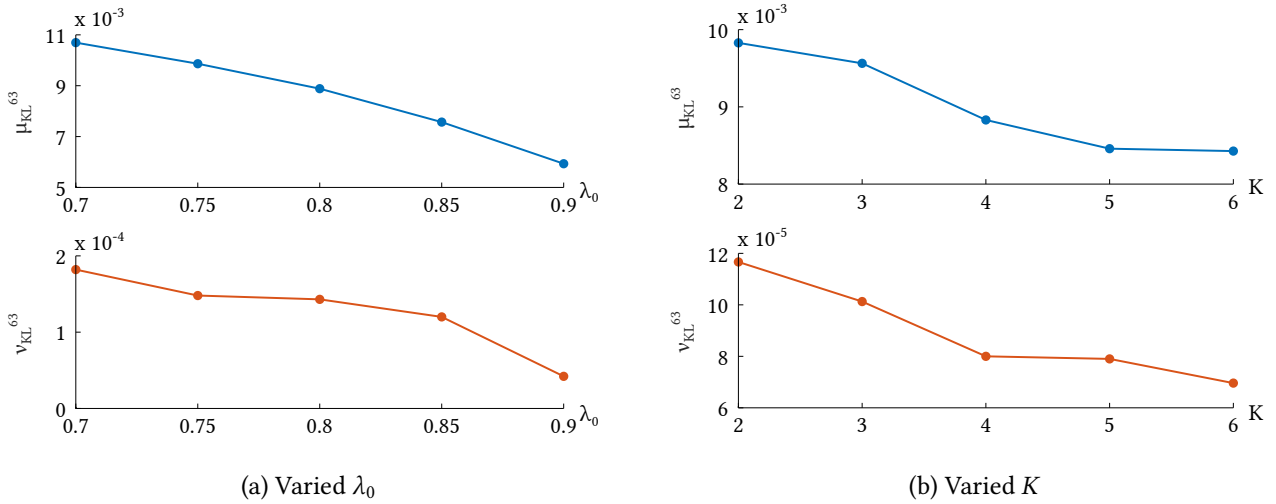


Figure 4.9: Evolution of μ_{KL}^{63} and v_{KL}^{63}

Similarly, we obtain in Figure 4.9b the evolution of μ_{KL}^{63} and v_{KL}^{63} when fixing $\lambda_0 = 0.75$ and $(QF_1, QF_2, QF_3) = (70, 80, 90)$ (i.e., $i = 3$), and varying K from 2 to 6 with step 1. This experiment allows to see the impact of K . Indeed, since the total portion of forged areas is always fixed at $1 - \lambda_0 = 0.25$, the higher value of K , the more the size of each forged area become smaller compared to the unmodified area. Therefore, the impact of forged areas on the statistical characteristics of AC DCT coefficients of tampered images is weaker, even they have the same portion in total (i.e., $1 - \lambda_0$). This allows to better approximate the mixture pdf $f_{X^{(1)}}(x; \Theta_K, \Lambda_K)$ by a single pdf $f_{X^{(1)}}(x; \hat{\theta})$, and hence a more accurate simplified model. As a result, μ_{KL}^{63} and v_{KL}^{63} slightly decrease in function of K as shown in Figure 4.9b.

Fixing now $K = 2$ and $\lambda_0 = 0.75$, we obtain in Table 4.2 the values of μ_{KL}^{63} and v_{KL}^{63} for all possible permutations of (QF_1, QF_2, QF_3) taken from the set $[70, 80, 90]$. Clearly, μ_{KL}^{63} and v_{KL}^{63} do not present a

QF_1	QF_2	QF_3	μ_{KL}^{63}	v_{KL}^{63}
70	90	80	0.005154	0.000212
90	70	80	0.006001	0.000156
80	90	70	0.006663	0.000219
70	80	90	0.009829	0.000117
80	70	90	0.013272	0.000159
90	80	70	0.015604	0.000231

Table 4.2: μ_{KL}^{63} and v_{KL}^{63} with respect to (QF_1, QF_2, QF_3)

tendency as in the two above case studies, however their small values still retain. This means that the simplified model (4.15) is robust to the quality factors.

4.4 Forgery Localization in Tampered JPEG Images

To show the utility of the proposed modeling framework, we apply the simplified model (4.15) to the forgery localization. Its performance is assessed by F_1 -score under divers configurations of tampered JPEG images.

4.4.1 Forgery Localization Method

From (4.15), the forgery localization returns to divide quantized AC DCT coefficients $\mathbf{y} = \{y_1, \dots, y_N\}$ of a tampered JPEG image into two clusters associated with the unmodified and forged areas. As recommended by McLachlan and Peel, the division can rely on the plug-in Bayes rule defined by [152, Section 1.15.2]

$$r_B \left(y_n; \hat{\boldsymbol{\theta}}, \hat{\lambda}_0, \hat{\mathbf{q}}^{(i)} \right) = \begin{cases} 1, & \text{if } y_n \text{ belongs to forged area,} \\ 0, & \text{otherwise,} \end{cases} \quad (4.20)$$

where $(\hat{\lambda}_0, \hat{\mathbf{q}}^{(i)})$ is the estimate of the parameter vector $(\lambda_0, \mathbf{q}^{(i)})$, and the condition “ y_n belongs to forged area” means $(1 - \hat{\lambda}_0) \cdot p_{Y^{(1)}}(y_n; \hat{\boldsymbol{\theta}}, \hat{\mathbf{q}}^{(i)}) \geq \hat{\lambda}_0 \cdot p_{Y^{(i)}}(y_n; \hat{\boldsymbol{\theta}}, \hat{\mathbf{q}}^{(i)})$ with $n = 1, \dots, N$. In practice, we can read the last quantization step $q^{(i)}$ from the JPEG header of the tampered image (by using e.g., Phil Sallee’s JPEG toolbox⁵), while the other parameters should be estimated from \mathbf{y} by maximizing the following log-likelihood function

$$\left(\hat{\lambda}_0, \hat{\mathbf{q}}^{(i-1)} \right) = \arg \max_{(\lambda_0, \mathbf{q}^{(i-1)})} \mathcal{L} \left(\lambda_0, \mathbf{q}^{(i-1)} \right), \quad (4.21)$$

where

$$\mathcal{L} \left(\lambda_0, \mathbf{q}^{(i-1)} \right) = \sum_{n=1}^N \ln \left(\lambda_0 \cdot p_{Y^{(i)}} \left(y_n; \hat{\boldsymbol{\theta}}, \mathbf{q}^{(i)} \right) + (1 - \lambda_0) \cdot p_{Y^{(1)}} \left(y_n; \hat{\boldsymbol{\theta}}, q^{(i)} \right) \right). \quad (4.22)$$

Moreover, since $q^{(1)}, \dots, q^{(i-1)}$ take discrete values within limited sets $\mathbf{Q}^{(1)}, \dots, \mathbf{Q}^{(i-1)}$, our estimation strategy is to fix, for each time, $\mathbf{q}^{(i-1)} = (q^{(1)}, \dots, q^{(i-1)})$ at given values $\tilde{\mathbf{q}}^{(i-1)} = (\tilde{q}^{(1)}, \dots, \tilde{q}^{(i-1)})$ of $\mathbf{Q}^{(1)}, \dots, \mathbf{Q}^{(i-1)}$, then to find $\tilde{\lambda}_0 = \arg \max_{\lambda_0} \mathcal{L} \left(\lambda_0, \tilde{\mathbf{q}}^{(i-1)} \right)$. The accepted estimates $(\hat{\lambda}_0, \hat{\mathbf{q}}^{(i-1)})$ is the vector $(\tilde{\lambda}_0, \tilde{\mathbf{q}}^{(i-1)})$ giving the highest log-likelihood $\mathcal{L} \left(\tilde{\lambda}_0, \tilde{\mathbf{q}}^{(i-1)} \right)$. This parameter estimation process can be done thanks to a simple EM algorithm. As this algorithm had been clearly presented in many textbooks (see e.g., [21] and [151]), we do not repeat its theoretical development here, but provide the pseudo-code in Algorithm 4.1 instead. We can assign $\tilde{\lambda}_0^{(0)}$ to a value in $(0.5, 1)$, and the stopping threshold $\delta_{\mathcal{L}}$ to a very small value (e.g. 10^{-6}).

For an illustration, we consider the tampered JPEG image in Figure 4.3b and take the GGD as the model of primary AC DCT coefficients. The associated binary localization maps with respect to 63 AC frequencies are shown in Figure 4.10. The white pixels correspond to forged area, while the black pixels associate with unmodified area. Of course, the localization performance can be improved

⁵The Phil Sallee’s JPEG toolbox can be downloaded from http://dde.binghamton.edu/download/jpeg_toolbox.zip

Algorithm 4.1 Pseudo-code of EM algorithm to estimate $(\lambda_0, \mathbf{q}^{(i-1)})$

Inputs: $\mathbf{y}, \delta_{\mathcal{L}}, \tilde{\lambda}_0^{(0)}, \mathbf{Q}^{(1)}, \dots, \mathbf{Q}^{(i-1)}$ ($\tilde{q}_{j_k}^{(k)} \in \mathbf{Q}^{(k)}, j_k = 1, \dots, J_k, k = 1, \dots, i-1$)

Outputs: $\hat{\lambda}_0, \tilde{\mathbf{q}}^{(i-1)}$

01. $\mathcal{L}_{\max} \leftarrow -\infty$
 02. **for** $j_1 = 1$ to J_1 **do**
 03. \vdots
 04. **for** $j_{i-1} = 1$ to J_{i-1} **do**
 05. $t \leftarrow 0, \tilde{\mathbf{q}}^{(i-1)} \leftarrow (\tilde{q}_{j_1}^{(1)}, \dots, \tilde{q}_{j_{i-1}}^{(i-1)})$
 06. $\mathcal{L}(\tilde{\lambda}_0^{(t)}, \tilde{\mathbf{q}}^{(i-1)}) \leftarrow \sum_{n=1}^N \ln \left(\tilde{\lambda}_0^{(t)} \cdot p_{Y^{(i)}}(y_n; \tilde{\boldsymbol{\theta}}, \tilde{\mathbf{q}}^{(i-1)}, q^{(i)}) + (1 - \tilde{\lambda}_0^{(t)}) \cdot p_{Y^{(1)}}(y_n; \tilde{\boldsymbol{\theta}}, q^{(i)}) \right)$
 07. **repeat**
 08. $t \leftarrow t + 1$
 09. // E-step //
 10. **for** $n = 1$ to N **do**
 11. $\omega^{(n)} \leftarrow \tilde{\lambda}_0^{(t)} \cdot p_{Y^{(i)}}(y_n; \tilde{\boldsymbol{\theta}}, \tilde{\mathbf{q}}^{(i-1)}, q^{(i)}) / \left(\tilde{\lambda}_0^{(t)} \cdot p_{Y^{(i)}}(y_n; \tilde{\boldsymbol{\theta}}, \tilde{\mathbf{q}}^{(i-1)}, q^{(i)}) + (1 - \tilde{\lambda}_0^{(t)}) \cdot p_{Y^{(1)}}(y_n; \tilde{\boldsymbol{\theta}}, q^{(i)}) \right)$
 12. **end for**
 13. // M-step //
 14. $\tilde{\lambda}_0^{(t)} \leftarrow \frac{1}{N} \sum_{n=1}^N \omega^{(n)}$
 15. $\mathcal{L}(\tilde{\lambda}_0^{(t)}, \tilde{\mathbf{q}}^{(i-1)}) \leftarrow \sum_{n=1}^N \ln \left(\tilde{\lambda}_0^{(t)} \cdot p_{Y^{(i)}}(y_n; \tilde{\boldsymbol{\theta}}, \tilde{\mathbf{q}}^{(i-1)}, q^{(i)}) + (1 - \tilde{\lambda}_0^{(t)}) \cdot p_{Y^{(1)}}(y_n; \tilde{\boldsymbol{\theta}}, q^{(i)}) \right)$
 16. **until** $\left| \mathcal{L}(\tilde{\lambda}_0^{(t+1)}, \tilde{\mathbf{q}}^{(i-1)}) - \mathcal{L}(\tilde{\lambda}_0^{(t)}, \tilde{\mathbf{q}}^{(i-1)}) \right| < \delta_{\mathcal{L}} \cdot \left| \mathcal{L}(\tilde{\lambda}_0^{(t+1)}, \tilde{\mathbf{q}}^{(i-1)}) \right|$
 17. **if** $\mathcal{L}(\tilde{\lambda}_0^{(t)}, \tilde{\mathbf{q}}^{(i-1)}) > \mathcal{L}_{\max}$ **then**
 18. $\mathcal{L}_{\max} \leftarrow \mathcal{L}(\tilde{\lambda}_0^{(t)}, \tilde{\mathbf{q}}^{(i-1)}), \hat{\lambda}_0 \leftarrow \tilde{\lambda}_0^{(t)}, \hat{\mathbf{q}}^{(i-1)} \leftarrow \tilde{\mathbf{q}}^{(i-1)}$
 19. **end if**
 20. **end for**
 21. \vdots
 22. **end for**
-

by applying one or more post-processings to these raw maps (see e.g., [217]). Still, to see better the intrinsic performance of the developed modeling framework for forgery localization, we decide not

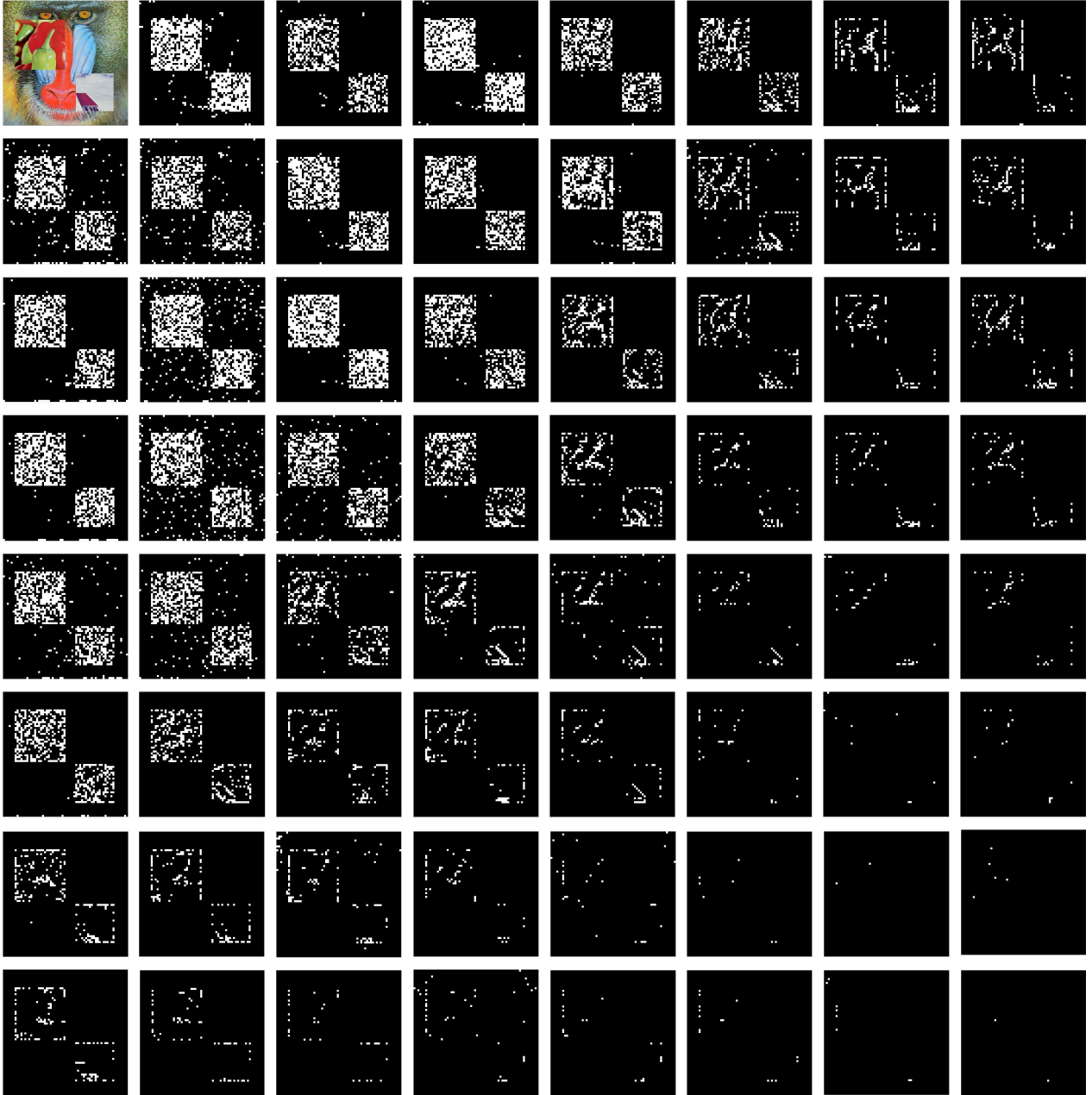


Figure 4.10: Binary localization maps with respect to 63 AC frequencies

to use the post-processing here. Visually, the localization performance is good at low or medium AC frequency, and becomes worse at higher frequency. This performance will be quantitatively studied in the next section.

4.4.2 Performance Assessment

We use F_1 -score to assess the forgery localization performance of the proposed modeling framework as in most related works (see e.g., [5]). Mathematically, it is computed by

$$F_1^{(i)} = \frac{2 \cdot TP^{(i)}}{2 \cdot TP^{(i)} + FN^{(i)} + FP^{(i)}}, \quad (4.23)$$

where i corresponds to the compression number of unmodified area, $TP^{(i)}$ denotes the number of tampered pixels correctly localized, $FN^{(i)}$ represents the number of unlocalized tampered pixels, and $FP^{(i)}$ stands for the number of authentic pixels wrongly localized. The higher the value of $F_1^{(i)}$, the more the forgery localization derived from $p_Y(y; \hat{\theta}, \lambda_0, \mathbf{q}^{(i)})$ is efficient. Using the GND as the model of primary unquantized AC DCT coefficients, we evaluate $F_1^{(i)}$, with $i = 2$ and 3 , on tampered JPEG images subject to *triple* quantization effect. As such, $i = 2$ and $i = 3$ correspond respectively to the wrong choice and right choice of model. Therefore, the comparison between $F_1^{(2)}$ and $F_1^{(3)}$ under various configurations of tampered JPEG images allows to quantify the gain returned by the right model, hence the advantage of the proposed framework. Besides, since the forgery localization is really effective in the range of low and medium AC frequencies (see Figure 4.10), we just evaluate the F_1 -scores at 10 frequencies therein.

Table 4.3 shows the values of $F_1^{(2)}$ and $F_1^{(3)}$ when fixing $K = 2$ and $(QF_1, QF_2, QF_3) = (70, 80, 90)$, and varying λ_0 from 0.7 to 0.9 with step 0.05. They decrease when λ_0 increases, because the size of forged areas becomes smaller, and hence harder to be localized. Similarly, we compute $F_1^{(2)}$ and $F_1^{(3)}$ when $\lambda_0 = 0.75$, $(QF_1, QF_2, QF_3) = (70, 80, 90)$, and K varies from 2 to 6 with step 1. Since we fix the portion of all forged areas at $1 - \lambda_0 = 0.25$, each of them becomes smaller for higher value of K , and therefore more difficult to be localized. This explains why both the $F_1^{(2)}$ and $F_1^{(3)}$ in Table 4.4 have a decreasing tendency when K increases. Now, we fix $\lambda_0 = 0.75$, $K = 2$, $(QF_1, QF_2) = (70, 80)$, vary QF_3 from 70 to 100 with step 5, and compute the associated $F_1^{(2)}$ and $F_1^{(3)}$. As shown in Table 4.5, for most considered AC frequencies, the F_1 -scores tend to increase with higher QF_3 . Especially, both the $F_1^{(2)}$ and $F_1^{(3)}$ are very small when $QF_3 \leq QF_2$. This is the inherent weakness of quantization effects-based methods [114], because the single and multiple compression artifacts in the histogram of AC DCT coefficients are no longer clearly distinguishable in such configurations.

In all the sensitivity studies of λ_0 , K and QF_3 , $F_1^{(3)}$ is greater than or at least equal to $F_1^{(2)}$. This confirms the usefulness of multiple JPEG compression modeling in the forgery localization. Indeed, considering the evolution of the difference $\Delta F_1 = F_1^{(3)} - F_1^{(2)}$ in Tables 4.3, 4.4 and 4.5, we obtain the highest gain at a small λ_0 , a small K and a not too high QF_3 .

4.5 Conclusion

We develop in this chapter a statistical modeling framework for DCT coefficients of tampered JPEG images where manipulations introduce single and multiple compression artifacts for forged and unmodified areas respectively. It allows to recursively derive a statistical model of DCT coefficients at any JPEG compression cycle from a known distribution of primary unquantized DCT coefficients. Through various numerical experiment and the K-L divergence, the framework is proved to be accurate if the distribution of primary unquantized AC DCT coefficients is either GND, GGD or CND. Especially, it exhibits its usefulness for forgery localization of tampered JPEG images subject to multiple quantization effect.

λ_0	f_{AC}	1	3	5	7	9	11	13	15	17	19
0.7	$F_1^{(3)}$	0.7277700	0.6940000	0.628916	0.689300	0.693600	0.665500	0.661100	0.501800	0.622600	0.617200
	$F_1^{(2)}$	0.622807	0.627009	0.628900	0.623615	0.632309	0.634536	0.612222	0.479681	0.574614	0.554067
	ΔF_1	0.104893	0.066991	0.000016	0.065685	0.061291	0.030964	0.048878	0.022119	0.047986	0.063133
0.75	$F_1^{(3)}$	0.715726	0.686105	0.618046	0.682158	0.681778	0.649118	0.652046	0.478870	0.614791	0.612307
	$F_1^{(2)}$	0.614772	0.623632	0.618046	0.620251	0.624128	0.624282	0.605098	0.468708	0.567845	0.553930
	ΔF_1	0.100954	0.062473	0.000000	0.061907	0.057650	0.024836	0.046948	0.010162	0.046946	0.058377
0.8	$F_1^{(3)}$	0.678600	0.664500	0.595700	0.653800	0.666800	0.615300	0.630000	0.424871	0.593100	0.583500
	$F_1^{(2)}$	0.586424	0.607982	0.595656	0.598433	0.616066	0.597684	0.589210	0.418000	0.548205	0.532850
	ΔF_1	0.092176	0.056518	0.000044	0.055367	0.050734	0.017616	0.040790	0.006871	0.044895	0.050650
0.85	$F_1^{(3)}$	0.654013	0.654034	0.586285	0.654005	0.652548	0.588893	0.632046	0.429355	0.593777	0.574376
	$F_1^{(2)}$	0.568088	0.605029	0.586285	0.604739	0.607059	0.579559	0.591710	0.426836	0.553137	0.536932
	ΔF_1	0.085925	0.049005	0.000000	0.049266	0.045489	0.009334	0.040336	0.002519	0.040640	0.037444
0.9	$F_1^{(3)}$	0.579900	0.592537	0.549542	0.613549	0.597903	0.5222930	0.594604	0.444288	0.557106	0.514438
	$F_1^{(2)}$	0.514106	0.566765	0.549542	0.584519	0.574440	0.516328	0.567462	0.442453	0.527294	0.502296
	ΔF_1	0.065794	0.025772	0.000000	0.029030	0.023463	0.006602	0.027142	0.001835	0.029812	0.012142

Table 4.3: F_1 -scores with respect to λ_0

K	f_{AC}	1	3	5	7	9	11	13	15	17	19
2	$F_1^{(3)}$	0.715726	0.686105	0.618046	0.682158	0.681778	0.649118	0.652046	0.47887	0.614791	0.612307
	$F_1^{(2)}$	0.614772	0.623632	0.618046	0.620251	0.624128	0.624282	0.605098	0.468708	0.567845	0.553930
	ΔF_1	0.100954	0.062473	0.000000	0.061907	0.057650	0.024836	0.046948	0.010162	0.046946	0.058377
3	$F_1^{(3)}$	0.704752	0.688542	0.624041	0.684130	0.689277	0.659332	0.661524	0.512978	0.627835	0.631608
	$F_1^{(2)}$	0.605262	0.622615	0.624041	0.619293	0.629023	0.630138	0.612769	0.500183	0.572156	0.564341
	ΔF_1	0.09949	0.065927	0.000000	0.064837	0.060254	0.029194	0.048755	0.127950	0.055679	0.067267
4	$F_1^{(3)}$	0.695917	0.670423	0.613475	0.668210	0.670266	0.632207	0.635828	0.483127	0.603520	0.594199
	$F_1^{(2)}$	0.597311	0.609245	0.613475	0.609379	0.614222	0.606994	0.588547	0.482786	0.556056	0.529182
	ΔF_1	0.098606	0.061178	0.000000	0.058831	0.056044	0.025213	0.047281	0.000341	0.047464	0.065017
5	$F_1^{(3)}$	0.687582	0.661760	0.597943	0.652348	0.650901	0.615633	0.618429	0.448440	0.580433	0.566797
	$F_1^{(2)}$	0.590068	0.605070	0.597943	0.595910	0.600973	0.596217	0.576682	0.448067	0.537210	0.509096
	ΔF_1	0.097514	0.056690	0.000000	0.056438	0.049928	0.019416	0.041747	0.000373	0.043223	0.057701
6	$F_1^{(3)}$	0.691600	0.663700	0.604214	0.647800	0.646100	0.608700	0.609900	0.435098	0.574500	0.571500
	$F_1^{(2)}$	0.594985	0.606479	0.604200	0.590000	0.590902	0.586059	0.570416	0.434100	0.536361	0.515406
	ΔF_1	0.096615	0.057221	0.000014	0.057800	0.055198	0.022641	0.039484	0.000998	0.038139	0.056094

Table 4.4: F_1 -scores with respect to K

QF_3	f_{AC}	1	3	5	7	9	11	13	15	17	19
70	$F_1^{(3)}$	0.385360	0.394117	0.350980	0.322175	0.382183	0.197404	0.240422	0.000297	0.000607	0.043737
	$F_1^{(2)}$	0.385360	0.379354	0.350980	0.280558	0.331113	0.192479	0.240408	0.000297	0.000548	0.043655
75	ΔF_1	0.000000	0.014763	0.000000	0.041617	0.051070	0.004925	0.000014	0.000000	0.000059	0.000082
	$F_1^{(3)}$	0.402361	0.320671	0.401035	0.299784	0.289505	0.324290	0.309725	0.290422	0.352355	0.297682
80	$F_1^{(2)}$	0.402361	0.320671	0.401030	0.299784	0.288768	0.307282	0.259880	0.251949	0.351489	0.231655
	ΔF_1	0.000000	0.000000	0.000005	0.000000	0.000737	0.017008	0.049845	0.038473	0.000866	0.066027
85	$F_1^{(3)}$	0.442586	0.310317	0.160921	0.293697	0.288087	0.277094	0.251629	0.157037	0.254497	0.257920
	$F_1^{(2)}$	0.442586	0.308961	0.129211	0.293570	0.282159	0.275690	0.248481	0.154022	0.253902	0.257135
90	ΔF_1	0.000000	0.001356	0.031710	0.000127	0.005928	0.001404	0.003148	0.003015	0.000595	0.000785
	$F_1^{(3)}$	0.565421	0.399742	0.336341	0.3399851	0.395517	0.405402	0.375440	0.277906	0.348291	0.325885
95	$F_1^{(2)}$	0.365574	0.260049	0.336206	0.244047	0.238800	0.301320	0.283815	0.121921	0.263775	0.223188
	ΔF_1	0.199847	0.139693	0.000135	0.155804	0.156717	0.104082	0.091625	0.155985	0.084516	0.102697
100	$F_1^{(3)}$	0.715726	0.686105	0.618046	0.682158	0.681778	0.649118	0.652046	0.478870	0.614791	0.612307
	$F_1^{(2)}$	0.614772	0.623632	0.618046	0.620251	0.624128	0.624282	0.605098	0.468708	0.567845	0.55393
95	ΔF_1	0.100954	0.062473	0.000000	0.061907	0.057650	0.024836	0.046948	0.010162	0.046946	0.058377
	$F_1^{(3)}$	0.709758	0.735266	0.680807	0.736079	0.731001	0.706584	0.802133	0.702995	0.801651	0.737790
100	$F_1^{(2)}$	0.672928	0.722453	0.680807	0.722344	0.718682	0.695373	0.786842	0.702184	0.791987	0.734050
	ΔF_1	0.036830	0.012813	0.000000	0.013735	0.012319	0.011211	0.015291	0.000811	0.009664	0.003740
100	$F_1^{(3)}$	0.709758	0.735262	0.680807	0.736069	0.731013	0.736223	0.746876	0.736497	0.738851	0.721200
	$F_1^{(2)}$	0.672928	0.722453	0.680807	0.722352	0.718682	0.735370	0.742758	0.732657	0.731501	0.718945
ΔF_1	0.036830	0.012809	0.000000	0.013717	0.012331	0.000853	0.004118	0.003840	0.007350	0.002255	

Table 4.5: F_1 -scores with respect to QF_3

Chapter 5

Statistical Detectors For Resampled TIFF Images

5.1 Introduction

In Chapter 3 and Chapter 4, we dealt with the tampering situations where *one or multiple parts* of an image have been manipulated. In this chapter, we consider that the *entire* image is falsified with malicious intent. For this kind of forgery, resampling (including resizing, rotation and other linear transformations) is a common technique [61]. It is usually used together with other tampering techniques such as cloning or splicing in order to create a visually perfect match for a forged image. However, no matter how sophisticated the resampling is, it always leaves two main traces similar to the ones left by the demosaicing (see Chapter 3): (i) the periodicity of resampled signal, and (ii) the incoherence in the variance between original and interpolated signals. As reviewed in Subsection 2.4.2.2 of Chapter 2, most authors have focused on the periodic artifacts to develop their resampling detectors. Additionally, statistical detectors based on the hypothesis testing theory are still very scattered irrespective of their both theoretical and practical interests. In front of this situation, we would like to take advantage of both the above traces to design in this chapter an optimal (generalized) likelihood ratio test (LRT)-based detector for resampled TIFF images. Compared to the current state-of-the-art (e.g., [139, 165, 181, 182]), the main innovations and contributions of our work are threefold.

1. Instead of using arbitrary probability distribution, we analyze the complete processing process from a RAW image to a resampled TIFF image to find out an appropriate statistical model for resampled TIFF images. This promises more accurate models for both the original and resampled TIFF images.
2. Based on the models developed for the original and resampled TIFF images, we exploit the periodic artifacts inherent in resampled pixels to simplify the models and extract independent identically distributed (IID) residual noises data for the LRT. The incoherence in the noises variance is next used as the main measure to distinguish a resampled TIFF image from an original one. This is the fundamental difference between our work and most related works whose resam-

pling detectors are based on the periodic artifacts. It results in two new (generalized) LRT-based detectors for the resampling detection.

3. Finally, as one of advantages of the hypothesis testing framework, we are able to provide closed-form expressions for the power function of the proposed detectors, and analytically analyze its properties.

We organize the remainder of this chapter as follows. Section 5.2 focuses on the mathematical analysis of resampled signals and on the statistical modeling of resampled TIFF images. In Section 5.3, we develop an ideal LRT-based detector for resampled TIFF images under the assumption that all the model parameters are known. The mathematical analysis of the LRT-based detector and its statistical performances is performed in detail here. Section 5.4 is dedicated to a more practical situation where all the model parameters, except the magnification rate, are assumed unknown. An estimation process is thus proposed to specify the unknown parameters before dealing with a generalized LRT (GLRT)-based detector. In Section 5.5, we carry out various numerical experiments on well-known image databases to validate the GLRT-based detector and to numerically assess its performances. Finally, some conclusions on the proposed (G)LRT-based detectors are discussed in Section 5.6.

5.2 Statistical Modeling of Resampled TIFF Images

To develop a model for resampled TIFF images, we propose to begin with a mathematical formulation of resampled signals and an analysis of their periodicity property. Next, we derive a statistical model for pixels of resampled TIFF images from by analyzing the complete processing process from a RAW image to a resampled TIFF image. For the sake of simplicity, we will restrict the study to single directional resampling.

5.2.1 Mathematical Analysis of Resampled Signals

Let $x(d) : \mathbb{R} \rightarrow \mathbb{R}$, denote an one-dimensional *continuous* real-valued signal¹ of interest in the spatial domain. We assume, without loss of generality, that $x(d)$ is uniformly sampled at positions $d = n \cdot \Delta$, with $\Delta = 1$, to produce the *discrete* real-valued signal $x_s[n] : \mathbb{Z} \rightarrow \mathbb{R}$, such that $x_s[n] = x(n \cdot \Delta) = x(n)$. Following [124, chapter 4], the resampling of $x_s[n]$ with factor $\xi = \frac{p}{q} > 0$ using interpolation kernel $h[\cdot]$ can be proceeded by three basic steps as in Figure 5.1.

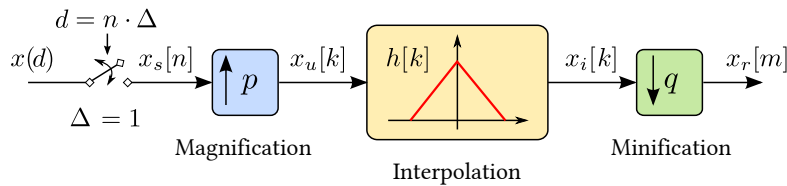


Figure 5.1: Block flow diagram of the resampling process

¹Continuous real-valued signal is one that has real intensity at every point in the real axis. The terms *continuous* should be understood as *everywhere defined* rather than in the mathematical sense.

1. *Magnification*: The sampled signal $x_s[n]$ is first expanded by rate p to create an upsampled signal

$$x_u[k] = \begin{cases} x_s[n] & \text{if } k = p \cdot n \\ 0 & \text{if } k \neq p \cdot n \end{cases}, \quad (5.1)$$

with $k \in \mathbb{Z}$.

2. *Interpolation*: The upsampled signal $x_u[k]$ is next convolved with the interpolation kernel $h[k]$ to generate an interpolated signal $x_i[k]$ at the same integer grid \mathbb{Z} such that

$$\begin{aligned} x_i[k] = x_u[k] * h[k] &= \sum_{j \in \mathbb{Z}} h[k-j] \cdot x_u[j] = \sum_{j \in \mathbb{Z}, j=p \cdot n} h[k-j] \cdot \underbrace{x_u[j]}_{x_s[n]} \\ &+ \sum_{j \in \mathbb{Z}, j \neq p \cdot n} h[k-j] \cdot \underbrace{x_u[j]}_0 = \sum_{n \in \mathbb{Z}} h[k-p \cdot n] \cdot x_s[n]. \end{aligned} \quad (5.2)$$

3. *Minification*. Finally, the interpolated signal $x_i[k]$ is compressed by rate q to obtain the resampled signal $x_r[m]$

$$x_r[m] = x_i[q \cdot m] = \sum_{n \in \mathbb{Z}} h[q \cdot m - p \cdot n] \cdot x_s[n], \quad (5.3)$$

with $m \in \mathbb{Z}$.

The magnification rate p and the minification rate q are constrained to integer [47]. Moreover, they are usually co-prime in practice (i.e., the greatest common divisor is 1). The resampling is called upsampling (by a fractional factor) if $\xi > 1$, and downsampling, otherwise.

Example 5.1. Figure 5.2 illustrates the resampling process with factor $\xi = \frac{p}{q} = \frac{4}{3}$ using *linear* interpolation kernel (see also Table 5.3 and Figure 5.9 for other kernels). As displayed in Figure 5.2b, the

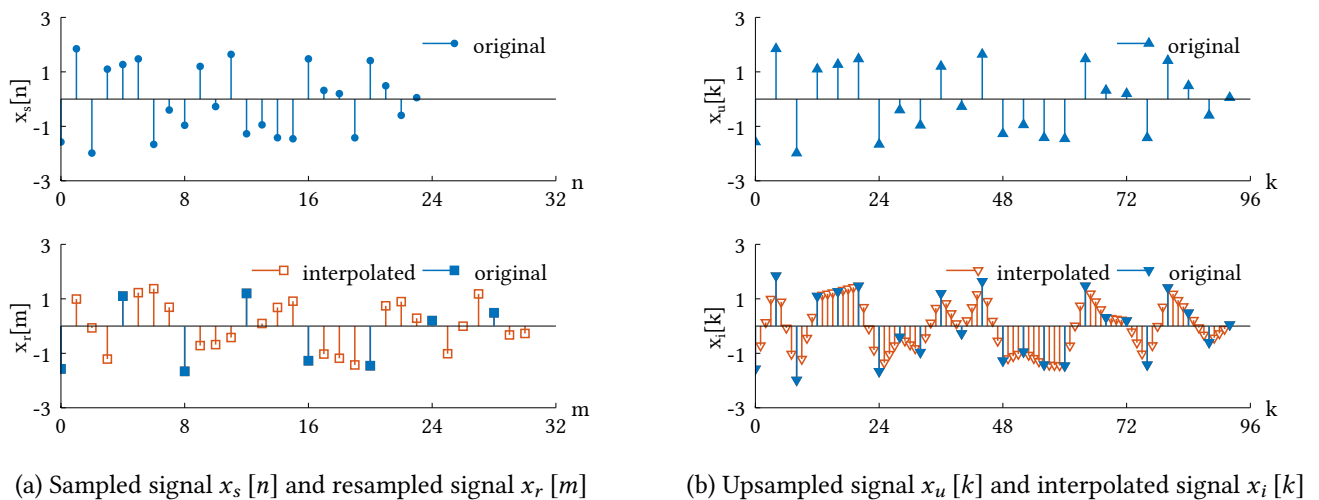


Figure 5.2: Illustration of resampling process with factor $\xi = \frac{4}{3}$ and linear interpolation kernel

interpolation operation allows to fill the missing positions due to the magnification by interpolated samples (∇) generated from the original ones (\blacktriangledown). After the minification, the resampled signal $x_r[m]$ consists of both the original samples (\blacksquare) at a lattice with period $p = 4$ since the first (original) sample and the interpolated samples (\square) at other positions (see Figure 5.2a). The length of $x_r[m]$ is equal to $\xi = \frac{p}{q} = \frac{4}{3}$ times of the one of $x_s[n]$.

Although the expression (5.3) favors the digitalization, it is not appropriate for the mathematical analysis owing to the different definition intervals of $x_r[m]$ and $x_s[n]$. To overcome this obstacle, a rescaling of m with factor $\omega = \xi^{-1} = \frac{q}{p}$ is proposed to obtain the same definition interval. As such, the above resampling process can be seen as a map from the integer grid \mathbb{Z} to a new discrete grid $\mathbb{Z}_\omega = \omega \cdot \mathbb{Z}$ through a continuous-time real-valued kernel $h(\delta) : \mathbb{R} \rightarrow \mathbb{R}$. Therefore, the values of the *rescaled* resampled signal $x_r(\omega \cdot m)$ are computed from the original signal $x(n)$ by [165]

$$x_r(\omega \cdot m) = \sum_{n \in \mathbb{Z}} h(\omega \cdot m - n) \cdot x(n), \quad (5.4)$$

with $m \in \mathbb{Z}$. The interpolation coefficient $h(\omega \cdot m - n)$ plays the role of weights in the linear combination (5.4). It has to verify the so-called *interpolation constraint* to not alter the value of original samples after the resampling

$$h(\omega \cdot m - n) = \begin{cases} 1 & \text{if } \omega \cdot m - n = 0, \\ 0 & \text{if } \omega \cdot m - n \in \mathbb{Z}^*. \end{cases} \quad (5.5)$$

It is also required that the interpolation reproduces constants, which is equivalent to [76]

$$\sum_{n \in \mathbb{Z}} h(\omega \cdot m - n) = 1 \quad \text{and} \quad |h(\omega \cdot m - n)| \leq 1, \quad \forall \omega \cdot m \in \mathbb{R} \quad (5.6)$$

Moreover, as specified by Gotchev *et al.* in [80], a good interpolation kernel is desired to be *symmetric* around 0 (i.e., $h(-\delta) = h(\delta)$) to avoid introducing phase distortions, and its support $[-S, S]$ (i.e., $h(\delta) = 0, \forall \delta \notin [-S, S]$) should be *as short as possible* to ensure the desired interpolation accuracy. The above constraints on $h(\omega \cdot m - n)$ lead us to rewrite (5.4) as

$$x_r(\omega \cdot m) = \sum_{n \in \mathcal{S}_h(m; \omega)} h(\omega \cdot m - n) \cdot x(n), \quad (5.7)$$

where $\mathcal{S}_h(m; \omega)$ denotes the set of indices n actually used in the interpolation

$$\begin{aligned} \mathcal{S}_h(m; \omega) &= \{n \in \mathbb{Z} \mid \omega \cdot m - n \in [-S, S]\} \\ &= \{\lceil \omega \cdot m - S \rceil, \lceil \omega \cdot m - S \rceil + 1, \dots, \lfloor \omega \cdot m + S \rfloor - 1, \lfloor \omega \cdot m + S \rfloor\}, \end{aligned} \quad (5.8)$$

where the notations $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ standard for rounds towards positive infinity and negative infinity respectively. We remark that

$$\mathcal{S}_h(m+p; \omega) = \mathcal{S}_h(m; \omega) + q, \quad (5.9)$$

so the resampled samples $x_r(\omega \cdot (m+p))$ can be obtained by combining original samples shifted by q positions with respect to the ones used to generate $x_r(\omega \cdot m)$.

Example 5.2. To illustrate the above rescaling process and the generation of the rescaled resampling signal using (5.7), let us continue with Example 5.1. The signal $x_r(\omega \cdot m)$ in Figure 5.3a is a scaled version of $x_r[m]$ in Figure 5.2a with scaling factor $\omega = \frac{q}{p} = \frac{3}{4}$. The interpolation coefficients $h(\cdot)$ are

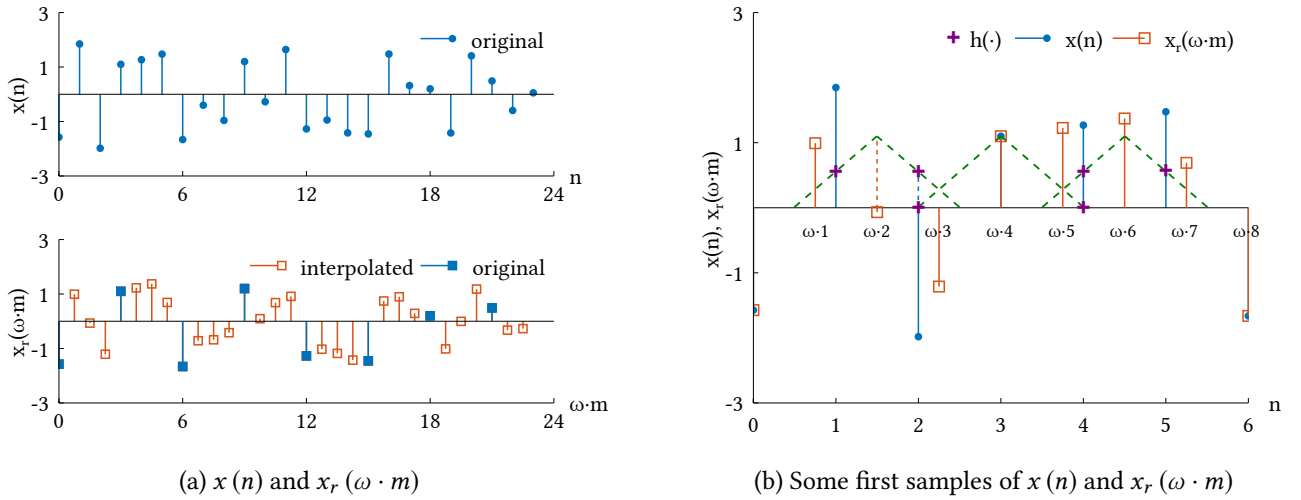


Figure 5.3: Illustration of original signal at discrete times $x(n)$ and rescaled resampled signal $x_r(\omega \cdot m)$

symbolized by \oplus in Figure 5.3b. We obtain the resampled sample $x_r(\omega \cdot 4)$ by combining the original samples $\{x(2), x(3), x(4)\}$ following the interpolation coefficients $\{0, 1, 0\}$. This actually returns an original sample (i.e., $x_r(\omega \cdot 4) \equiv x(3)$). Similarly, $x_r(\omega \cdot 2)$ and $x_r(\omega \cdot 6)$ are interpolated from the sets $\{x(1), x(2)\}$ and $\{x(4), x(5)\}$ respectively. Clearly, their indices satisfy (5.9).

Hereinafter, we analyze some useful properties of the rescaled resampled signal in the spatial domain.

Theorem 5.1. *If the sample $x_r(\omega \cdot m)$ of the rescaled resampled signal with scaling factor $\omega = \xi^{-1} = \frac{q}{p}$ is an original sample, then the samples $x_r(\omega \cdot (m + j \cdot p)), \forall j \in \mathbb{Z}$, are also original samples.*

Proof. From the interpolation constraint (5.5), a sample of the rescaled resampled signal is an original sample if and only if it is at integer grid \mathbb{Z} . So, it is enough to verify that $\omega \cdot (m + j \cdot p) \in \mathbb{Z}$ when $\omega \cdot m \in \mathbb{Z}$. Indeed, since q and $j \in \mathbb{Z}$, it is evident that $\omega \cdot (m + j \cdot p) = \omega \cdot m + j \cdot q \in \mathbb{Z}$. \square

Theorem 5.1 means that the original samples of a rescaled resampled signal with scaling factor $\omega = \xi^{-1} = \frac{q}{p}$ are at lattice of period p . For instance, the original samples of a rescaled resampled signal in Figure 5.3b are at positions $0, \omega \cdot 4, \omega \cdot 8, \dots$ as $p = 4$.

Theorem 5.2. *All samples $x_r(\omega \cdot (m + j \cdot p)), \forall j \in \mathbb{Z}$, of the rescaled resampled signal with scaling factor $\omega = \xi^{-1} = \frac{q}{p}$ share the same set of interpolation coefficients.*

Proof. The above statement implies that the interpolation coefficients used in (5.7) are repeated with period p . Let $\mathcal{H}_h(m; \omega) = \{h(\omega \cdot m - n) \mid n \in \mathcal{S}_h(m; \omega)\}$ be set of interpolation coefficients used to generate $x_r(\omega \cdot m)$, we will show that

$$\mathcal{H}_h(m + p; \omega) = \mathcal{H}_h(m; \omega). \quad (5.10)$$

Indeed, from the definition of $\mathcal{H}_h(m; \omega)$, we obtain

$$\mathcal{H}_h(m + p; \omega) = \{h(\omega \cdot (m + p) - n) \mid n \in \mathcal{S}_h(m + p; \omega)\} \stackrel{\omega = \frac{q}{p}}{=} \{h(\omega \cdot m + q - n) \mid n \in \mathcal{S}_h(m + p; \omega)\}. \quad (5.11)$$

Using (5.9), we rewrite (5.11) as

$$\begin{aligned} \mathcal{H}_h(m + p; \omega) &= \{h(\omega \cdot m + q - n) \mid n \in \mathcal{S}_h(m; \omega) + q\} \\ &= \{h(\omega \cdot m - (n - q)) \mid n - q \in \mathcal{S}_h(m; \omega)\} = \mathcal{H}_h(m; \omega). \end{aligned} \quad (5.12)$$

Another proof of the periodicity of $\mathcal{H}_h(m; \omega)$ can be found in [108]. \square

In Figure 5.3b, the sets of interpolation coefficients to compute $x_r(\omega \cdot 2)$ and $x_r(\omega \cdot 6) = x_r(\omega \cdot (2 + 4))$ are the same.

Now, assuming that $x(n)$ is independently Gaussian distributed with expectation $\mu_{x(n)}$ and variance $\sigma_{x(n)}^2$ (i.e., $x(n) \sim \mathcal{N}(\mu_{x(n)}, \sigma_{x(n)}^2)$), then from (5.7), $x_r(\omega \cdot m)$ is also Gaussian distributed

$$x_r(\omega \cdot m) \sim \mathcal{N}(\mu_{x_r(\omega \cdot m)}, \sigma_{x_r(\omega \cdot m)}^2), \quad (5.13)$$

with expectation

$$\mu_{x_r(\omega \cdot m)} = \sum_{n \in \mathcal{S}_h(m, \omega)} h(\omega \cdot m - n) \cdot \mu_{x(n)}, \quad (5.14)$$

and variance

$$\sigma_{x_r(\omega \cdot m)}^2 = \sum_{n \in \mathcal{S}_h(m, \omega)} h^2(\omega \cdot m - n) \cdot \sigma_{x(n)}^2. \quad (5.15)$$

The statistical distribution (5.13) is the basis to build the statistical model for pixels of resampled TIFF images.

5.2.2 Statistical Model for Resampled TIFF Pixels

A typical process to create a resampled TIFF image from a real scene is depicted in Figure 5.4. Firstly, light intensity measured at each pixel generates an electrical signal that is read out as a RAW image. Next, the RAW image goes through several in-camera post-acquisition processes, such as *demosaicing*, *white balancing* and *gamma correction*, to issue a full resolution, colored and uncompressed image called *original* TIFF image. Finally, for malicious purposes, *resampling* is applied to the original TIFF image to create *resampled* TIFF image. Our goal is to develop a statistical model for resampled TIFF

pixels by step-by-step analyzing processing stages from the RAW image to the resampled TIFF image.

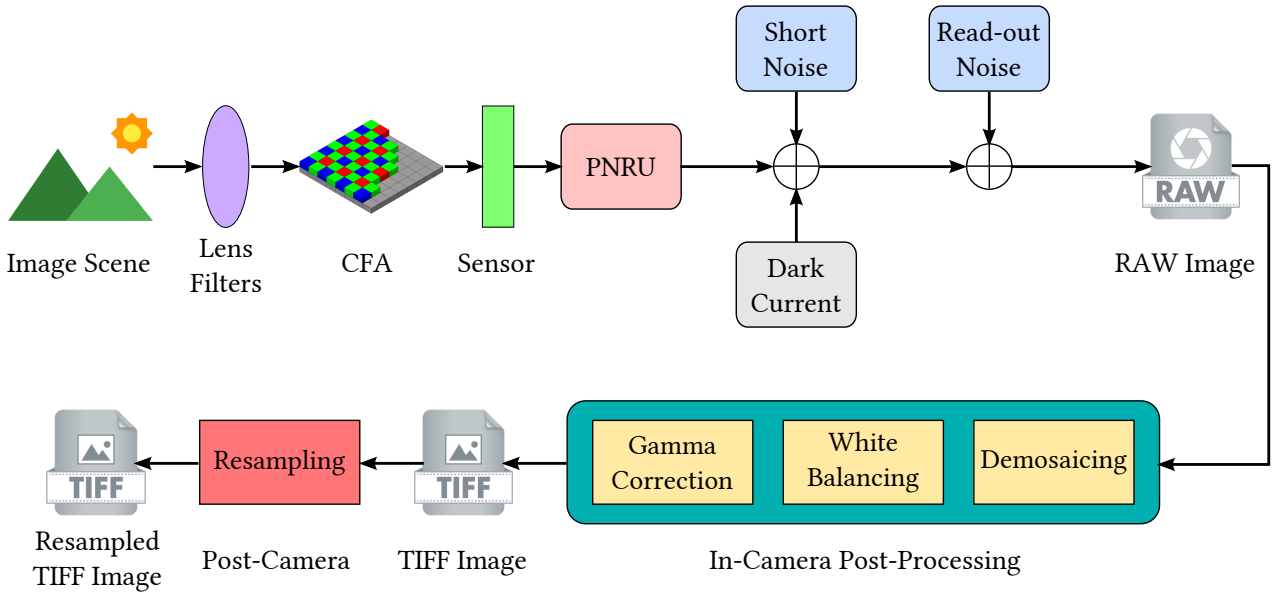


Figure 5.4: Illustration of process to create resampled TIFF images

Let denote v_n the pixel located at position n of a RAW image². Due to the stochastic nature of photon noise and other disturbances (see Figure 5.4), v_n can be expressed as

$$v_n = \mu_{v_n} + \epsilon_{v_n}, \quad (5.16)$$

where μ_{v_n} denotes the expectation of v_n in the absence of noise, and ϵ_{v_n} stands for all noise sources that interfere with the original signal. A typical representation of ϵ_{v_n} is a *Poissonian-Gaussian noise model* [141], where the Poissonian part characterizes signal-dependent component including dark current and shot noise, while the Gaussian part describes signal-independent component such as read-out noise. For practical applications, Foi *et al.* [67] propose to simplify the Poissonian-Gaussian noise model by treating the Poisson noise as a special heteroscedastic Gaussian. This leads to the so-called *heteroscedastic noise model* [205] of RAW pixels

$$v_n \sim \mathcal{N}(\mu_{v_n}, \tilde{a} \cdot \mu_{v_n} + \tilde{b}). \quad (5.17)$$

This model describes the pixel noise variance as an affine function of the pixel expectation, parameterized by the couple (\tilde{a}, \tilde{b}) . Since the noise corrupting each RAW pixel is independent of those of neighbor pixels [86], RAW pixels are also mutually independent.

To establish a model for *original* TIFF pixels from the heteroscedastic noise model (5.17), the demosaicing and white balancing are assumed to be linear [186]. This allows to retain the Gaussian

²For the sake of simplicity, we use single index rather than the couple of row and column indices to specify the location of a pixel in an image.

distribution for independent pixels³ after these operations, but with different parameters (a, b)

$$w_n \sim \mathcal{N}(\mu_{w_n}, a \cdot \mu_{w_n} + b), \quad (5.18)$$

where w_n denotes a white-balanced pixel at position n , and μ_{w_n} is its expectation. The gamma correction is, on the contrary, a non-linear operation defined by the following pixel-wise power-law [59]

$$x_n = w_n^{\frac{1}{\gamma}}, \quad (5.19)$$

where γ is the correction factor, and x_n denotes the pixel at position n of the original TIFF image. The non-linear expression (5.19) of x_n does not allow a simple statistical distribution, thus we seek its linear approximation. To this end, we first rewrite (5.18) as

$$w_n = \mu_{w_n} + \epsilon_{w_n}, \quad \text{with} \quad \epsilon_{w_n} \sim \mathcal{N}(0, a \cdot \mu_{w_n} + b), \quad (5.20)$$

where ϵ_{w_n} accounts for zero-mean signal-dependent Gaussian noise after the white balancing. This leads to

$$x_n = (\mu_{w_n} + \epsilon_{w_n})^{\frac{1}{\gamma}} = \mu_{w_n}^{\frac{1}{\gamma}} \cdot \left(1 + \frac{\epsilon_{w_n}}{\mu_{w_n}}\right)^{\frac{1}{\gamma}}. \quad (5.21)$$

Using now the first order of Taylor series expansion of $(1+t)^{\frac{1}{\gamma}}$ at $t=0$, we finally arrive at

$$x_n = \mu_{w_n}^{\frac{1}{\gamma}} \cdot \left(1 + \frac{1}{\gamma} \cdot \frac{\epsilon_{w_n}}{\mu_{w_n}} + o\left(\frac{\epsilon_{w_n}}{\mu_{w_n}}\right)\right) \simeq \mu_{w_n}^{\frac{1}{\gamma}} + \frac{1}{\gamma} \cdot \mu_{w_n}^{\frac{1}{\gamma}-1} \cdot \epsilon_{w_n}. \quad (5.22)$$

Since $\epsilon_{w_n} \sim \mathcal{N}(0, a \cdot \mu_{w_n} + b)$, it follows that

$$x_n \sim \mathcal{N}\left(\mu_{w_n}^{\frac{1}{\gamma}}, \frac{1}{\gamma^2} \cdot \mu_{w_n}^{\frac{2}{\gamma}-2} \cdot (a \cdot \mu_{w_n} + b)\right). \quad (5.23)$$

Let denote $\mu_{x_n} = \mu_{w_n}^{\frac{1}{\gamma}}$ the expectation of x_n , then from (5.23), its variance $\sigma_{x_n}^2$ be expressed as

$$\sigma_{x_n}^2 = \frac{1}{\gamma^2} \cdot \mu_{x_n}^{2-2\gamma} \cdot (a \cdot \mu_{x_n}^{\gamma} + b) \triangleq f(\mu_{x_n}; a, b, \gamma), \quad (5.24)$$

and original TIFF pixels are independently Gaussian distributed such that

$$x_n \sim \mathcal{N}(\mu_{x_n}, \sigma_{x_n}^2). \quad (5.25)$$

The expression (5.24) is known as *generalized signal-dependent noise model* in [207]. Given an original TIFF image, the triple of parameters (a, b, γ) can be estimated using the classical maximum likelihood (ML) method (see also Subsection 5.4.1). In [208], Thai *et al.* observe through numerical experiments

³Although the image pixels are actually no longer independent after the demosaicing (see Chapter 3), we still assume their statistical independence to facilitate the modeling.

that the ML estimates (\hat{a}, \hat{b}) of (a, b) are invariant to image scenes and camera settings, but discriminative for different camera models, while the ML estimate $\hat{\gamma}$ of γ is almost the same irrespective to the diversity of camera models.

We seek now a statistical model of the *resampled* TIFF pixels. A resampling operation with factor $\xi = \frac{p}{q}$ and interpolation kernel $h(\cdot)$ is considered. To further simplify, we assume that the resampling has been performed in one direction (either horizontal or vertical) overall the original TIFF image. Given the independent Gaussian distribution (5.25) of original TIFF pixels x_n , with $n \in \mathcal{S}_h(m, \omega)$ and $\omega = \xi^{-1} = \frac{q}{p}$, we can apply (5.13) to characterize the statistics of resampled TIFF pixel y_m at position m as

$$y_m \sim \mathcal{N}(\mu_{y_m}, \sigma_{y_m}^2), \quad (5.26)$$

where the expectation μ_{y_m} and variance $\sigma_{y_m}^2$ are given by

$$\mu_{y_m} \triangleq e(\mu_{x_n}, m, n; h, \omega) = \sum_{n \in \mathcal{S}_h(m, \omega)} h(\omega \cdot m - n) \cdot \mu_{x_n}, \quad (5.27)$$

and

$$\sigma_{y_m}^2 \triangleq \vartheta(\mu_{x_n}, m, n; h, \omega, a, b, \gamma) = \sum_{n \in \mathcal{S}_h(m, \omega)} h^2(\omega \cdot m - n) \cdot f(\mu_{x_n}; a, b, \gamma). \quad (5.28)$$

Obviously, μ_{y_m} and $\sigma_{y_m}^2$ depend not only on the expectation of original TIFF pixels μ_{x_n} , but also on the interpolation coefficients $h(\omega \cdot m - n)$ defined by the relative distances between the considered pixel and their neighbors $\omega \cdot m - n$. Applying the constraints (5.6) to (5.27) and (5.28), we find that

$$\mu_{y_m} \leq \mu_{x_n} \quad \text{and} \quad \sigma_{y_m}^2 \leq \sigma_{x_n}^2, \quad (5.29)$$

which make sense because the interpolation step in the resampling process smooths the image, hence the expectation and variance of resampled TIFF pixels are smaller than the ones of original TIFF pixels.

5.3 Ideal Detector for Resampled TIFF Images

Our goal is to develop in this section an efficient detector within statistical hypothesis testing framework to distinguish an original TIFF image from a resampled one. Naturally, the statistical models (5.25) and (5.26) of original and resampled TIFF pixels could be used. However, these models may lead to difficulties in mathematical formulation because they are not suitable to independent identically distributed (IID) pixels. To bypass this obstacle, we propose a preprocessing stage to obtain IID datasets associated with a simplified version of (5.25) and (5.26). Subsequently, we develop an optimal *likelihood ratio test* (LRT)-based detector under the assumption that all model parameters are known. Its statistical performance is also analytically evaluated.

5.3.1 Analysis of TIFF Pixels Models

Let consider the model (5.25) of original TIFF pixels x_n , we find that $x_n \sim \mathcal{N}(\mu_{x_n}, \sigma_{x_n}^2)$ are independent but not identically distributed because of their scene-dependent expectation μ_{x_n} . As the variance $\sigma_{x_n}^2 = f(\mu_{x_n}; a, b, \gamma)$ is a function of μ_{x_n} , with $f(\cdot)$ given from (5.24), it enough to set μ_{x_n} as a constant to make these pixels identically distributed. This idea leads to the *homogeneous block detection* and *level-set segmentation* process described in Subsection 5.3.2. Indeed, this process allows to partition the considered original TIFF image into $I \in \mathbb{N}$ non-overlapping segments of $J_i \in \mathbb{N}$ pixels for the segment S_i , $i \in \{1, \dots, I\}$. In each segment S_i , all the pixels expectations $\mu_{x_{i,j}}$, $j \in \{1, \dots, J_i\}$, are almost the same and near to the local expectation $\mu_{x_{i,j}} \approx \mu_i$, and hence almost the same variance $\sigma_i^2 = f(\mu_i; a, b, \gamma)$. As a result, all the pixels $x_{i,j}$ of S_i are IID as

$$x_{i,j} \sim \mathcal{N}(\mu_i, f(\mu_i; a, b, \gamma)), \quad (5.30)$$

where $i \in \{1, \dots, I\}$ and $j \in \{1, \dots, J_i\}$ denote respectively the index of the segment S_i and of the j -th pixel therein.

If the same homogeneous block detection and level-set segmentation process as above is applied to the considered resampled TIFF image, we obtain from (5.27), (5.28) and (5.6) the expectation and variance of pixels $y_{i,j}$, $i \in \{1, \dots, I\}$ and $j \in \{1, \dots, J_i\}$, in a non-overlapping homogeneous segment S_i as

$$e(\mu_i, m, n; h, \omega) = \sum_{n \in \mathcal{S}_h(m, \omega)} h(\omega \cdot m - n) \cdot \mu_i = \mu_i \cdot \sum_{n \in \mathcal{S}_h(m, \omega)} h(\omega \cdot m - n) = \mu_i \quad (5.31)$$

and

$$\begin{aligned} \vartheta(\mu_i, m, n; h, \omega, a, b, \gamma) &= \sum_{n \in \mathcal{S}_h(m, \omega)} h^2(\omega \cdot m - n) \cdot f(\mu_i; a, b, \gamma) \\ &= f(\mu_i; a, b, \gamma) \cdot \sum_{n \in \mathcal{S}_h(m, \omega)} h^2(\omega \cdot m - n) = \frac{1}{\gamma^2} \cdot \mu_i^{2-2\gamma} \cdot (a \cdot \mu_i^\gamma + b) \cdot \sum_{n \in \mathcal{S}_h(m, \omega)} h^2(\omega \cdot m - n) \\ &= f(\mu_i; A(m, n; a, h, \omega), B(m, n; b, h, \omega), \gamma), \end{aligned} \quad (5.32)$$

where

$$A(m, n; a, h, \omega) = a \cdot \sum_{n \in \mathcal{S}_h(m, \omega)} h^2(\omega \cdot m - n) \text{ and } B(m, n; b, h, \omega) = b \cdot \sum_{n \in \mathcal{S}_h(m, \omega)} h^2(\omega \cdot m - n). \quad (5.33)$$

Since there are different sets of interpolation coefficients on a resampled TIFF image, we can derive from (5.32) that their pixels $y_{i,j}$ remain non-identical. Luckily, as stated in Theorem 5.2, the interpolation coefficients are repeated with period p such that $\mathcal{H}_h(m; \omega) = \mathcal{H}_h(m + p; \omega)$. So, we can extract sub-images at lattice of period p from the resampled TIFF image in order that the sets of interpolation coefficients on each sub-image become identical and $\sum_{n \in \mathcal{S}_h(m, \omega)} h^2(\omega \cdot m - n) = \zeta_u$ for the u -th sub-image. By this way, after applying the homogeneous segmentation technique to a given sub-image u , we obtain

$$e(\mu_i, m, n; h, \omega) = \mu_i \quad \text{and} \quad \vartheta(\mu_i, m, n; h, \omega, a, b, \gamma) = f(\mu_i; a \cdot \zeta_u, b \cdot \zeta_u, \gamma), \quad (5.34)$$

and the resampled TIFF pixels $y_{i,j}$ over a segment S_i , $i \in \{1, \dots, I\}$ and $j \in \{1, \dots, J_i\}$, become IID as

$$y_{i,j} \sim \mathcal{N}(\mu_i, f(\mu_i; a \cdot \zeta_u, b \cdot \zeta_u, \gamma)). \quad (5.35)$$

Clearly, by limiting TIFF images to a lattice of period p and to non-overlapping homogeneous segments, pixels of the original and resampled sub-images have the same mean but not the same variance. Actually, their variances share the same form, but since $\zeta_u = \sum_{n \in \mathcal{S}_h(m, \omega)} h^2(\omega \cdot m - n) \leq 1$, the one of resampled TIFF pixels is smaller. This remark implies that the pixels variance could be a good measure to differentiate between original and resampled TIFF images.

5.3.2 IID Data Extraction

The above analyses on pixels models allow us to build IID datasets of noisy pixels, denoised pixels and residual noises from an unknown (i.e., either original or resampled) TIFF image as in Figure 5.5. It consists of three main consecutive steps: (i) *sub-images extraction*, (ii) *homogeneous block detection*,

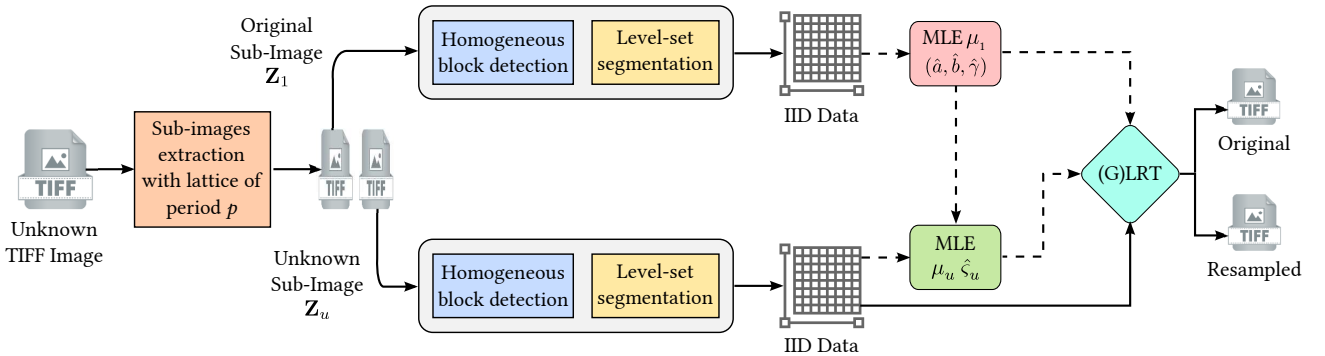


Figure 5.5: Illustration of process for IID data extraction from an unknown TIFF image

and (iii) *level-set segmentation*.

Let denote U a two-dimensional matrix representing a color channel (i.e., red, green or blue) of an unknown TIFF image, we decompose U following lattices with period p to obtain p sub-images $Z_u = U(u : p : \text{end})^4$, with $u \in \{1, \dots, p\}$. If the unknown TIFF image is the original one, all the sub-images Z_u , $\forall u \in \{1, \dots, p\}$, are also original; otherwise, only the first sub-image Z_1 is original following Theorem 5.1, and the others Z_u , $\forall u \in \{2, \dots, p\}$ are interpolated⁵. The original sub-image Z_1 will be used to estimate the parameters of image and camera model. Meanwhile, the other sub-images will be exploited to estimate the sum of squared interpolation coefficients, and to be used as input of the (generalized) LRT-based detectors. These points will be treated more detail in Section 5.4.

⁴This notation is borrowed from the MathWorks' Matlab language.

⁵For the sake of simplification, we assume that the first line or the first row of the unknown TIFF image is the original one.

Edge or discontinuity are the main factors that cause the high fluctuation from pixel to pixel in a sub-image \mathbf{Z}_u , $u \in \{1, \dots, p\}$. To reduce this variation, we only consider the parts of \mathbf{Z}_u at which no local edge and discontinuity exist. To identify them, we first apply a denoising filter \mathcal{D} to decompose the sub-image \mathbf{Z}_u into an approximate image structure (i.e., denoised sub-image) $\mathbf{T}_u = \mathcal{D}(\mathbf{Z}_u)$ and the associated residue $\mathbf{R}_u = \mathbf{Z}_u - \mathbf{T}_u$. Although several denoising filters such as *Gaussian*, *Wiener*, *BM3D* [48] could be applied, we just use the *wavelet-based filter* introduced in [154] because it gives the best compromise between the accuracy and computational efficiency. Considering the sub-image \mathbf{Z}_u as N_b image blocks of size 8×8^6 , we next divide \mathbf{Z}_u into 64 vectors of pixels $\mathbf{z}_{u,l} = (z_{u,l,1}, \dots, z_{u,l,N_b})$, where $l \in \{1, \dots, 64\}$ denotes the location index in the 8×8 grid of image blocks. All the pixels in the same 8×8 block $L \in \{1, \dots, N_b\}$ is represented by the vector $\mathbf{z}_u^L = (z_{u,1,L}, \dots, z_{u,64,L})$. The same process is applied on \mathbf{T}_u and \mathbf{R}_u to respectively obtain the vector $\mathbf{t}_u^L = (t_{u,1,L}, \dots, t_{u,64,L})$ and $\mathbf{r}_u^L = (r_{u,1,L}, \dots, r_{u,64,L})$ for each 8×8 block $L \in \{1, \dots, N_b\}$. A block L is seen as homogeneous if the associated standard deviation σ_u^L computed from \mathbf{t}_u^L is smaller than a threshold τ_u . In [207], Thai *et al.* propose to compute the standard deviation σ_u^L in the DCT domain as

$$\sigma_u^L = 1.4826 \cdot \text{MAD}(\text{DCT}(t_{u,1,L}, \dots, t_{u,64,L})), \quad (5.36)$$

and take the threshold τ_u as the median of absolute deviations of all pixels in the residual sub-image \mathbf{R}_u

$$\tau_u = 1.4826 \cdot \text{MAD}(\mathbf{R}_u), \quad (5.37)$$

where $\text{DCT}(\cdot)$ stands for the *discrete cosine transform* and $\text{MAD}(\cdot)$ denotes the *median of absolute deviations* [190]. Consequently, the set of homogeneous blocks of the sub-image \mathbf{Z}_u is defined by

$$B_u = \{L \in \{1, \dots, N_b\} : \sigma_u^L \leq \tau_u\}, \quad (5.38)$$

where σ_u^L and τ_u are given from (5.36) and (5.37) respectively.

Once the homogeneous block detection done, it is proposed to use one or multiple of 64 vectors of pixels $\mathbf{z}_{u,l}$, with $l \in \{1, \dots, 64\}$, for partitioning into I non-overlapping segments $S_{u,i}$, $i \in \{1, \dots, I\}$ with width Δ_i . Each homogeneous segment $S_{u,i}$ is defined by

$$S_{u,i} = \left\{ \mathbf{z}_{u,l,L} \mid t_{u,l,L} \in \left[n_i - \frac{\Delta_i}{2}, n_i + \frac{\Delta_i}{2} \right], L \in B_u \right\}. \quad (5.39)$$

The smaller the value of Δ_i , the more the IID property of pixels in homogeneous segments is ensured, but the less the number of pixels is gathered. In practice, the choice of $I = 2^8$ for 8-bits images and $\Delta_i = 1$ usually gives the best compromise between the quality and the quantity of IID pixels.

As a result, for an unknown TIFF image, we get a set of $J_{u,i}$ IID pixels denoted by $\mathbf{z}_{u,i} = \{z_{u,i,j}\}_{j=1}^{J_{u,i}}$ in each segment $S_{u,i}$, with $i \in \{1, \dots, I\}$, satisfying

⁶Other size of image blocks is possible. The choice of 8×8 image blocks is inspired from the fact that JPEG compression works separately on 8×8 image blocks.

$$z_{u,i,j} \sim \mathcal{N}\left(\mu_{u,i}, \sigma_{u,i}^2\right), \quad (5.40)$$

where $\mu_{u,i}$ and $\sigma_{u,i}^2 = f(\mu_{u,i}; a_u, b_u, \gamma)$ are respectively the local expectation and local variance of pixels over $S_{u,i}$. The parameters a_u and b_u are expressed by

$$a_u = \zeta_u \cdot a \quad \text{and} \quad b_u = \zeta_u \cdot b, \quad (5.41)$$

with ζ_u the sum of squared interpolation coefficients for the sub-image Z_u defined by

$$\zeta_u = \begin{cases} 1 & \text{if } Z_u \text{ is an original sub-image,} \\ \sum_{n \in S_h(m, \omega)} h^2(\omega \cdot m - n) & \text{if } Z_u \text{ is an interpolated sub-image.} \end{cases} \quad (5.42)$$

Associated with $z_{u,i}$, we can also obtain the IID sets of denoised pixels $\mathbf{t}_{u,i} = \{t_{u,i,j}\}_{j=1}^{J_{u,i}}$ and residual noises $\mathbf{r}_{u,i} = \{r_{u,i,j}\}_{j=1}^{J_{u,i}}$. It is easy to derive from (5.39) that $t_{u,i,j}$, $j \in \{1, \dots, J_{u,i}\}$, are uniformly distributed

$$t_{u,i,j} \sim \mathcal{U}\left(n_i - \frac{\Delta_i}{2}, n_i + \frac{\Delta_i}{2}\right). \quad (5.43)$$

Meanwhile, since $r_{u,i,j} = z_{u,i,j} - t_{u,i,j}$, $j \in \{1, \dots, J_{u,i}\}$, the *probability density function* (pdf) of $r_{u,i,j}$ can be obtained by

$$f_{r_{u,i,j}}(r) = \frac{1}{\Delta_i} \cdot \left(\Phi\left(\frac{n_i + \frac{\Delta_i}{2} + r - \mu_{u,i}}{\sigma_{u,i}}\right) - \Phi\left(\frac{n_i - \frac{\Delta_i}{2} + r - \mu_{u,i}}{\sigma_{u,i}}\right) \right), \quad (5.44)$$

where $\Phi(\cdot)$ stands for the standard normal *cumulative distribution function* (cdf). Moreover, as we are able to choose Δ_i small enough in order that the local expectation $\mu_{u,i}$ almost equals n_i , the pdf $f_{r_{u,i,j}}(r)$ becomes

$$f_{r_{u,i,j}}(r) \xrightarrow{\Delta_i \rightarrow 0} \frac{1}{\sigma_{u,i}} \cdot \varphi\left(\frac{r}{\sigma_{u,i}}\right), \quad (5.45)$$

where $\varphi(\cdot)$ denotes the standard normal pdf. In other words, with an appropriate choice of Δ_i , the population of residual noises can be well approximated by a zero-mean Gaussian distribution with variance $\sigma_{u,i}^2$

$$r_{u,i,j} \sim \mathcal{N}\left(0, \sigma_{u,i}^2\right). \quad (5.46)$$

Although three kinds of IID datasets and the associated models are available, we decide to use only \mathbf{t}_u and \mathbf{r}_u hereafter, because of the incoherence between the discrete integer value of pixels $z_{u,i,j}$ and their continuous Gaussian distribution $\mathcal{N}(\mu_{u,i}, \sigma_{u,i}^2)$ with $\mu_{u,i}$ and $\sigma_{u,i}^2 \in \mathbb{R}$.

5.3.3 LRT-Based Detector and Statistical Performance

Given the IID residual noise data $\mathbf{r}_u \triangleq \{r_{u,i,j}\}$, $\forall i \in \{1, \dots, I\}$ and $j \in \{1, \dots, J_{u,i}\}$ from an unknown TIFF sub-image Z_u , our aim is to design an optimal detector to check if the considered TIFF image is original or resampled in an ideal context where all model parameters are known. We first formulate

such a detection problem as a binary hypothesis testing, then we rely on the LRT for two simple hypotheses to propose an optimal detector and evaluate its statistical performance.

From \mathbf{r}_u and the model (5.46), we can formulate the detection problem as the decision between two following hypotheses

$$\begin{cases} \mathcal{H}_0 = \left\{ r_{u,i,j} \sim \mathcal{N}\left(0, \sigma_{0,u,i}^2\right), \forall i \in \{1, \dots, I\}, \forall j \in \{1, \dots, J_{u,i}\} \right\} : \text{original TIFF image,} \\ \mathcal{H}_1 = \left\{ r_{u,i,j} \sim \mathcal{N}\left(0, \sigma_{1,u,i}^2\right), \forall i \in \{1, \dots, I\}, \forall j \in \{1, \dots, J_{u,i}\} \right\} : \text{resampled TIFF image,} \end{cases} \quad (5.47)$$

where $u \in \{2, \dots, p\}$ is the index of the tested sub-image \mathbf{Z}_u , $\sigma_{0,u,i}^2$ and $\sigma_{1,u,i}^2$ are the pixels variances with respect to $\mu_{u,i}$ under the hypotheses \mathcal{H}_0 and \mathcal{H}_1

$$\sigma_{0,u,i}^2 = f(\mu_{u,i}; a, b, \gamma), \quad (5.48)$$

and

$$\sigma_{1,u,i}^2 = f(\mu_{u,i}; a_u, b_u, \gamma), \quad (5.49)$$

with $f(\cdot)$, a_u and b_u given from (5.24) and (5.41). We are interested in a test δ that guarantees a prefixed false-alarm rate α_0 such that

$$\mathcal{K}_{\alpha_0} = \left\{ \delta : P_{\mathcal{H}_0}(\delta(\mathbf{r}_u) = \mathcal{H}_1) \leq \alpha_0 \right\}. \quad (5.50)$$

Among all the tests in the class \mathcal{K}_{α_0} , we look for the most powerful test δ^* satisfying

$$\delta^* = \arg \max_{\delta \in \mathcal{K}_{\alpha_0}} \beta(\delta), \quad (5.51)$$

where $\beta(\delta)$ is power function defined by

$$\beta(\delta) = P_{\mathcal{H}_1}(\delta(\mathbf{r}_u) = \mathcal{H}_1). \quad (5.52)$$

We assume that the magnification rate p , the vector of image parameters $\boldsymbol{\mu}_u = (\mu_{u,1}, \dots, \mu_{u,i}, \dots, \mu_{u,I})$, the triplet of camera parameters (a, b, γ) , and the sum of squared interpolation coefficients ζ_u are known in advance. Under such an ideal context, the Neyman-Pearson fundamental lemma [129, theorem 3.2.1] specifies that the most powerful test δ^* for the problem (5.47) under the LRT defined by the decision rule

$$\delta^*(\mathbf{r}_u) = \begin{cases} \mathcal{H}_0 & \text{if } \Lambda(\mathbf{r}_u) = \sum_{i=1}^I \sum_{j=1}^{J_{u,i}} \Lambda(r_{u,i,j}) < \theta_u \\ \mathcal{H}_1 & \text{if } \Lambda(\mathbf{r}_u) = \sum_{i=1}^I \sum_{j=1}^{J_{u,i}} \Lambda(r_{u,i,j}) \geq \theta_u \end{cases}, \quad (5.53)$$

where $\Lambda(\mathbf{r}_u)$ is log-likelihood ratio (LLR) of \mathbf{r}_u under the ideal context, θ_u denotes the decision threshold which is the solution of the equation

$$P_{\mathcal{H}_0}(\Lambda(\mathbf{r}_u) \geq \theta_u) = \alpha_0, \quad (5.54)$$

and $\Lambda(r_{u,i,j})$ denotes the LLR of one sample $r_{u,i,j}$ computed by

$$\Lambda(r_{u,i,j}) = \frac{1}{2} \cdot \left(\frac{\sigma_{1,u,i}^2}{\sigma_{0,u,i}^2} - 1 \right) \cdot \frac{r_{u,i,j}^2}{\sigma_{1,u,i}^2} - \frac{1}{2} \cdot \log \left(\frac{\sigma_{1,u,i}^2}{\sigma_{0,u,i}^2} \right). \quad (5.55)$$

Since $\frac{\sigma_{1,u,i}^2}{\sigma_{0,u,i}^2} = \zeta_u$, we can further express $\Lambda(r_{u,i,j})$ as

$$\Lambda(z_{u,i,j}) = \frac{1}{2} \cdot \left(1 - \frac{1}{\zeta_u} \right) \cdot \frac{r_{u,i,j}^2}{\sigma_{0,u,i}^2} - \frac{1}{2} \cdot \log \zeta_u. \quad (5.56)$$

To determine the decision threshold θ_u from (5.54), as well as the performance of LRT-based detector, the survival function $\bar{F}_{\Lambda(\mathbf{r}_u), \mathcal{H}_k}(\theta_u)$ of $\Lambda(\mathbf{r}_u)$ under the hypothesis \mathcal{H}_k , $k \in \{0, 1\}$ is required. Proposition 5.1 provides the closed-form expression of $\bar{F}_{\Lambda(\mathbf{r}_u), \mathcal{H}_k}(\theta_u)$.

Proposition 5.1. *Under the hypothesis \mathcal{H}_k , $k \in \{0, 1\}$, the survival function of $\Lambda(\mathbf{r}_u)$ is computed by*

$$\bar{F}_{\Lambda(\mathbf{r}_u), \mathcal{H}_k}(\theta_u) = \begin{cases} G\left(\rho_u \cdot \log \frac{1}{\zeta_u} - \theta_u; \rho_u, \frac{1}{\zeta_u} - 1\right) & \text{if } k = 0, \\ G\left(\rho_u \cdot \log \frac{1}{\zeta_u} - \theta_u; \rho_u, 1 - \zeta_u\right) & \text{if } k = 1, \end{cases} \quad (5.57)$$

where $G(\cdot)$ stands for the Gamma cdf defined by the following regularized incomplete gamma function

$$G(x; \eta, \lambda) = \frac{\Gamma(\eta, \frac{x}{\lambda})}{\Gamma(\eta)}, \quad (5.58)$$

with $\Gamma(\eta) = \int_0^\infty s^{\eta-1} e^{-s} ds$ and $\Gamma(\eta, x) = \int_0^x s^{\eta-1} e^{-s} ds$ the gamma function and the lower incomplete gamma function respectively, and ρ_u denotes the half of total residual noises considered for the sub-image \mathbf{Z}_u

$$\rho_u = \frac{1}{2} \cdot \sum_{i=1}^I J_{u,i}. \quad (5.59)$$

Proof. Using (5.56), we rewrite the LLR $\Lambda(\mathbf{r}_u)$ as follow

$$\Lambda(\mathbf{r}_u) = \sum_{i=1}^I \sum_{j=1}^{J_{u,i}} \Lambda(r_{u,i,j}) = -R_u + c_u, \quad (5.60)$$

where the random variable R_u and the constant c are respectively defined by

$$R_u = \frac{1}{2} \cdot \left(\frac{1}{\zeta_u} - 1 \right) \cdot \sum_{i=1}^I \left(\frac{1}{\sigma_{0,u,i}^2} \cdot \sum_{j=1}^{J_{u,i}} r_{u,i,j}^2 \right), \quad (5.61)$$

and

$$c_u = \rho_u \cdot \log \frac{1}{\zeta_u}. \quad (5.62)$$

To derive $\bar{F}_{\Lambda(\mathbf{r}_u), \mathcal{H}_k}(\theta_u)$ from (5.60), we should determine the distribution of R_u under the hypothesis

\mathcal{H}_k first. Indeed, under the hypothesis \mathcal{H}_k , $k \in \{0, 1\}$, $r_{u,i,j} \sim \mathcal{N}(0, \sigma_{k,u,i}^2)$, hence $\frac{r_{u,i,j}}{\sigma_{k,u,i}} \sim \mathcal{N}(0, 1)$. It follows that [99, Chapter 18]

$$\left(\frac{r_{u,i,j}}{\sigma_{k,u,i}}\right)^2 \sim \chi_1^2, \quad (5.63)$$

where χ_1^2 stands for the chi-square probability distribution with one degree of freedom. Therefore

$$r_{u,i,j}^2 \sim \mathcal{G}\left(\frac{1}{2}, 2 \cdot \sigma_{k,u,i}^2\right), \quad (5.64)$$

where $\mathcal{G}(\eta, \lambda)$ denotes the Gamma probability distribution with pdf

$$g(x; \eta, \lambda) = \frac{1}{\lambda^\eta \cdot \Gamma(\eta)} \cdot x^{\eta-1} \cdot e^{-\frac{x}{\lambda}}. \quad (5.65)$$

Using (5.61) and (5.64), we arrive at

$$R_u \sim \mathcal{G}(\rho_u, \xi_{u,k}), \quad (5.66)$$

where

$$\xi_{u,k} = \begin{cases} \frac{1}{\zeta_u} - 1 & \text{if } k = 0: \text{ under } \mathcal{H}_0, \\ 1 - \zeta_u & \text{if } k = 1: \text{ under } \mathcal{H}_1. \end{cases} \quad (5.67)$$

We can therefore compute $\bar{F}_{\Lambda(\mathbf{r}_u), \mathcal{H}_k}(\theta_u)$ by

$$\bar{F}_{\Lambda(\mathbf{r}_u), \mathcal{H}_k}(\theta_u) = P_{\mathcal{H}_k}(\Lambda(\mathbf{r}_u) \geq \theta_u) = P_{\mathcal{H}_k}(R_u \leq c_u - \theta_u) = G(c_u - \theta_u; \rho_u, \xi_{u,k}), \quad (5.68)$$

with $G(c_u - \theta_u; \rho_u, \xi_{u,k})$ derived from (5.58). Substituting (5.62) and (5.67) into (5.68), the result in Proposition 5.1 follows immediately. \square

We can therefore summary the LRT-based detector $\delta^*(\mathbf{r}_u)$ under the ideal context by the following corollary.

Corollary 5.1. *When all the model parameters are known, the optimal detector to distinguish an original TIFF image from a resampled TIFF image is as follows*

$$\delta^*(\mathbf{r}_u) = \begin{cases} \mathcal{H}_0 : \text{ original TIFF image} & \text{if } \Lambda(\mathbf{r}_u) < \theta_u, \\ \mathcal{H}_1 : \text{ resampled TIFF image} & \text{if } \Lambda(\mathbf{r}_u) \geq \theta_u, \end{cases} \quad (5.69)$$

where $u \in \{2, \dots, p\}$, $\Lambda(\mathbf{r}_u)$ is given by

$$\Lambda(\mathbf{r}_u) = -\frac{1}{2} \cdot \left(\frac{1}{\zeta_u} - 1\right) \cdot \sum_{i=1}^I \left(\frac{1}{\sigma_{0,u,i}^2} \cdot \sum_{j=1}^{J_{u,i}} r_{u,i,j}^2\right) + \rho_u \cdot \log \frac{1}{\zeta_u} \quad (5.70)$$

and θ_u is computed as

$$\theta_u = \bar{F}_{\Lambda(\mathbf{r}_u), \mathcal{H}_0}^{-1}(\alpha_0), \quad (5.71)$$

in which $\bar{F}_{\Lambda(\mathbf{r}_u), \mathcal{H}_0}^{-1}(\cdot)$ is inverse function of $\bar{F}_{\Lambda(\mathbf{r}_u), \mathcal{H}_0}(\cdot)$ in (5.57)

$$\bar{F}_{\Lambda(\mathbf{r}_u), \mathcal{H}_0}(\theta_u) = G\left(\rho_u \cdot \log \frac{1}{\zeta_u} - \theta_u; \rho_u, \frac{1}{\zeta_u} - 1\right), \quad (5.72)$$

with $G(\cdot)$ and ρ_u given from (5.58) and (5.59) respectively.

From (5.52) and (5.69), we can express the power function of $\delta^*(\mathbf{r}_u)$ as

$$\beta(\delta^*(\mathbf{r}_u)) = P_{\mathcal{H}_1}(\delta^*(\mathbf{r}_u) = \mathcal{H}_1) = P_{\mathcal{H}_1}(\Lambda(\mathbf{r}_u) \geq \theta_u). \quad (5.73)$$

Using (5.57), we arrive at

$$\beta(\delta^*(\mathbf{r}_u)) = G\left(\rho_u \cdot \log \frac{1}{\zeta_u} - \theta_u; \rho_u, 1 - \zeta_u\right), \quad (5.74)$$

where θ_u is computed following (5.71), $G(\cdot)$ is given by (5.58), and ρ_u is obtained by (5.59). We remark from (5.74) that $\beta(\delta^*(\mathbf{r}_u))$ depends only on ζ_u and ρ_u of the tested sub-image \mathbf{Z}_u , $u = 2, \dots, p$.

Proposition 5.2. *The power function defined by (5.74) is non-decreasing in ρ_u and non-increasing in ζ_u .*

Proof. Using (5.58), we express (5.74) in the form

$$\beta(\delta^*(\mathbf{r}_u)) = \frac{\Gamma\left(\rho_u, \frac{-\rho_u \cdot \log \zeta_u - \theta}{1 - \zeta_u}\right)}{\Gamma(\rho_u)} \triangleq Q(\zeta_u, \rho_u). \quad (5.75)$$

Applying the Corollary 3.1 in [72] to $Q(\zeta_u, \rho_u)$, we find directly that $\beta(\delta^*(\mathbf{r}_u))$ is non-decreasing in ρ_u . Taking now the derivative of $Q(\zeta_u, \rho_u)$ with respect to ζ_u , we obtain

$$\frac{\partial Q(\zeta_u, \rho_u)}{\partial \zeta_u} = -\frac{1}{\Gamma(\rho_u)} \cdot e^{\frac{\rho_u \cdot \log \zeta_u + \theta}{1 - \zeta_u}} \cdot \left(\frac{-\rho_u \cdot \log \zeta_u - \theta}{1 - \zeta_u}\right)^{\rho_u - 1} \cdot \frac{\frac{\rho_u}{\zeta_u} \cdot (1 - \zeta_u) + \rho_u \cdot \log \zeta_u + \theta}{(1 - \zeta_u)^2}. \quad (5.76)$$

Since $\frac{\partial Q(\zeta_u, \rho_u)}{\partial \zeta_u} \leq 0$, $\forall \zeta_u \in [0, 1]$, the power function $\beta(\delta^*(\mathbf{r}_u))$ is non-increasing in ζ_u . \square

5.4 Practical Detector for Resampled TIFF Images

Despite its theoretical interest, the ideal LRT-based detector designed in the previous section is very constrained in practical applications because the model parameters are usually unknown. This section aims to relax these constraints to facilitate the applications in practice by considering that the magnification rate p is the only known parameter⁷. This leads to the decision between two composite hypotheses (5.47) which is commonly treated by an optimal test under large conditions, namely *generalized likelihood ratio test* (GLRT) [228]. The concept of GLRT is very similar to one of LRT, but the unknown parameters in the expression (5.70) of $\Lambda(\mathbf{r}_u)$ are replaced by their ML estimates. Therefore, we have to

⁷The magnification rate p can be estimated with high precision by using the Qiao *et al.*'s method [181].

propose an estimation process to specify the vector of image parameters $\boldsymbol{\mu}_u = (\mu_{u,1}, \dots, \mu_{u,i}, \dots, \mu_{u,I})$, the triplet of camera parameters (a, b, γ) , as well as the sum of squared interpolation coefficients ζ_u before theoretically dealing with the GLRT and its performance.

5.4.1 Estimation of Model Parameters

We start with the estimation of the local expectation $\mu_{u,i}$ of each homogeneous segment $S_{u,i}$. From the homogeneous block detection and level-set segmentation process described in Subsection 5.3.2, it is easy to see that the sample mean of denoised pixels $t_{u,i,j}$ is a good estimate of $\mu_{u,i}$

$$\hat{\mu}_{u,i} = \frac{1}{J_{u,i}} \cdot \sum_{j=1}^{J_{u,i}} t_{u,i,j}. \quad (5.77)$$

Indeed, since the variance of the denoised pixels $t_{u,i,j}$ is almost negligible [207] (see also Subsection 5.5), $\hat{\mu}_{u,i}$ is very near to the true value $\mu_{u,i}$. This explains why $\hat{\mu}_{u,i}$ is treated as a constant in the following developments.

We propose to use the data extracted from the *original* sub-image Z_1 to estimate (a, b, γ) , because the relationship between the local variance $\sigma_{1,i}^2$ and the local expectation $\mu_{1,i}$ of pixels over a segment $S_{1,i}$, $i \in \{1, \dots, I\}$ is invariant for whatever the real nature (i.e., original or resampled) of the considered TIFF image U

$$\sigma_{1,i}^2 = f(\mu_{1,i}; a, b, \gamma), \quad (5.78)$$

where $f(\cdot)$ is given from (5.24). An estimate $(\hat{a}, \hat{b}, \hat{\gamma})$ of (a, b, γ) can be therefore obtained by fitting $f(\mu_{1,i}; a, b, \gamma)$ to the scatter-plot of couples $\{\mu_{1,i}, \sigma_{1,i}^2\}_{i=1}^I$. However, since $\mu_{1,i}$ and $\sigma_{1,i}^2$ are unknown in practice, we could use their estimate $\hat{\mu}_{1,i}$ and $\hat{\sigma}_{1,i}^2$ instead, where $\hat{\mu}_{1,i}$ is the sample mean given from (5.77), and $\hat{\sigma}_{1,i}^2$ is the *unbiased* sample variance of residues $r_{1,i,j}$, $j \in \{1, \dots, J_{1,i}\}$,

$$\hat{\sigma}_{1,i}^2 = \frac{1}{J_{1,i} - 1} \cdot \sum_{j=1}^{J_{1,i}} (r_{1,i,j} - \bar{r}_{1,i,j}), \quad \text{with } \bar{r}_{1,i,j} = \frac{1}{J_{1,i}} \sum_{j=1}^{J_{1,i}} r_{1,i,j}. \quad (5.79)$$

While $\hat{\mu}_{1,i}$ can be merely treated as a constant (i.e., $\hat{\mu}_{1,i} \approx \mu_{1,i}$), $\hat{\sigma}_{1,i}^2$ is more crucial and should be carefully studied. Indeed, since $r_{1,i,j} \sim \mathcal{N}(0, \sigma_{1,i}^2)$, it follows from (5.79) that [111, Proposition 2.11]

$$\frac{J_{1,i} - 1}{\sigma_{1,i}^2} \cdot \hat{\sigma}_{1,i}^2 \sim \chi_{J_{1,i}-1}^2 \quad (5.80)$$

where $\chi_{J_{1,i}-1}^2$ denotes chi-square distribution with $J_{1,i} - 1$ degrees of freedom. Hence,

$$\hat{\sigma}_{1,i}^2 \sim \mathcal{G}\left(\frac{J_{1,i} - 1}{2}, \frac{2}{J_{1,i} - 1} \cdot \sigma_{1,i}^2\right) = \mathcal{G}\left(\frac{J_{1,i} - 1}{2}, \frac{2}{J_{1,i} - 1} \cdot f(\hat{\mu}_{1,i}; a, b, \gamma)\right), \quad (5.81)$$

where $\mathcal{G}(\eta, \lambda)$ is the Gamma statistical distribution. Fitting now $f(\hat{\mu}_{1,i}; a, b, \gamma)$ to the scatter-plot

$\{\hat{\mu}_{1,i}, \hat{\sigma}_{1,i}^2\}_{i=1}^I$ by the ML approach, we obtain an estimate of (a, b, γ) as follows

$$\left(\hat{a}, \hat{b}, \hat{\gamma}\right) = \arg \max_{(a,b,\gamma)} \mathcal{L}(a, b, \gamma), \quad (5.82)$$

where $\mathcal{L}(a, b, \gamma)$ is the log-likelihood function of $\hat{\sigma}_{1,i}^2$ over I segments $S_{1,i}$ given by

$$\begin{aligned} \mathcal{L}(a, b, \gamma) &= \sum_{i=1}^I \log g\left(\hat{\sigma}_{1,i}^2; \frac{J_{1,i}-1}{2}, \frac{2}{J_{1,i}-1} \cdot f(\hat{\mu}_{1,i}; a, b, \gamma)\right) = \sum_{i=1}^I \left(\frac{J_{1,i}-3}{2} \cdot \log \hat{\sigma}_{1,i}^2 - \log \Gamma\left(\frac{J_{1,i}-1}{2}\right)\right. \\ &\quad \left. - \frac{J_{1,i}-1}{2} \cdot \log \frac{2}{J_{1,i}-1} - \frac{J_{1,i}-1}{2} \cdot \log f(\hat{\mu}_{1,i}; a, b, \gamma) - \frac{J_{1,i}-1}{2} \cdot \frac{1}{f(\hat{\mu}_{1,i}; a, b, \gamma)} \hat{\sigma}_{1,i}^2\right). \end{aligned} \quad (5.83)$$

The complexity of $\mathcal{L}(a, b, \gamma)$ does not allow an analytical solution of (5.82), we should therefore resort to numerical optimization methods for this goal. More precisely, removing constant parts with respect to a, b and γ from (5.83), we obtain

$$\left(\hat{a}, \hat{b}, \hat{\gamma}\right) = \arg \min_{(a,b,\gamma)} \sum_{i=1}^I \left((J_{1,i}-1) \cdot \left(\log f(\hat{\mu}_{1,i}; a, b, \gamma) + \frac{\hat{\sigma}_{1,i}^2}{f(\hat{\mu}_{1,i}; a, b, \gamma)} \right) \right). \quad (5.84)$$

We next apply the Nelder-Mead simplex algorithm [120] (i.e., the `fminsearch` function in Matlab) to (5.84) to search $(\hat{a}, \hat{b}, \hat{\gamma})$. According to [111, Theorem 5.4], the ML estimates \hat{a}, \hat{b} and $\hat{\gamma}$ are asymptotically consistent, i.e., they asymptotically converge in probability to their true value

$$\left(\hat{a}, \hat{b}, \hat{\gamma}\right) \xrightarrow{p} (a, b, \gamma). \quad (5.85)$$

Although the number of considered IID pixels for a sub-image \mathbf{Z}_u , $u \in \{1, \dots, p\}$, is actually finite, it is usually large enough to ensure a high accuracy of the ML estimates. This observation has been confirmed by various numerical experiments synthetic images in [207, Table 1]. In the following developments, we will treat $\hat{a}, \hat{b}, \hat{\gamma}$ as constants to further simplify, even though they intrinsically exhibit a certain variability.

We seek now the estimate of the sum of square interpolation coefficients ζ_u associated with a sub-image \mathbf{Z}_u , $u \in \{2, \dots, p\}$. From (5.32), the local variance $\sigma_{u,i}^2$ of pixels over a segment $S_{u,i}$, $i \in \{1, \dots, I\}$, of \mathbf{Z}_u is expressed as

$$\sigma_{u,i}^2 = f(\mu_{u,i}; \zeta_u \cdot a, \zeta_u \cdot b, \gamma) = \zeta_u \cdot f(\mu_{u,i}; a, b, \gamma), \quad (5.86)$$

with $\zeta_u \leq 1$. An estimate of ζ_u can be found by fitting $f(\mu_{u,i}; \zeta_u \cdot a, \zeta_u \cdot b, \gamma) = \zeta_u \cdot f(\mu_{u,i}; a, b, \gamma)$ to the scatter-plot of $\{\hat{\mu}_{u,i}, \hat{\sigma}_{u,i}^2\}_{i=1}^I$, where $\hat{\mu}_{u,i}$ is an estimate of $\mu_{u,i}$ given from (5.77) and $\hat{\sigma}_{u,i}^2$ an estimate

of $\sigma_{u,i}^2$ defined by

$$\hat{\sigma}_{u,i}^2 = \frac{1}{J_{u,i} - 1} \cdot \sum_{j=1}^{J_{u,i}} (r_{u,i,j} - \bar{r}_{u,i,j}), \quad \text{with } \bar{r}_{u,i,j} = \frac{1}{J_{u,i}} \sum_{j=1}^{J_{u,i}} r_{u,i,j}. \quad (5.87)$$

Since $\hat{\sigma}_{u,i}^2$ is Gamma distributed

$$\hat{\sigma}_{u,i}^2 \sim \mathcal{G} \left(\frac{J_{u,i} - 1}{2}, \frac{2}{J_{u,i} - 1} \cdot \zeta_u \cdot f(\hat{\mu}_{u,i}; a, b, \gamma) \right), \quad (5.88)$$

the ML estimate $\hat{\zeta}_u$ of ζ_u is given by

$$\hat{\zeta}_u = \arg \max_{\zeta_u} \mathcal{L}(\zeta_u), \quad (5.89)$$

with $\mathcal{L}(\zeta_u)$ the log-likelihood function obtained in a similar way to (5.83) as

$$\begin{aligned} \mathcal{L}(\zeta_u) = \sum_{i=1}^I & \left(\frac{J_{u,i} - 3}{2} \cdot \log \hat{\sigma}_{u,i}^2 - \log \Gamma \left(\frac{J_{u,i} - 1}{2} \right) - \frac{J_{u,i} - 1}{2} \cdot \log \frac{2}{J_{u,i} - 1} \right. \\ & \left. - \frac{J_{u,i} - 1}{2} \cdot \log f(\hat{\mu}_{u,i}; a, b, \gamma) - \frac{J_{u,i} - 1}{2} \cdot \log \zeta_u - \frac{J_{u,i} - 1}{2} \cdot \frac{1}{\zeta_u \cdot f(\hat{\mu}_{u,i}; a, b, \gamma)} \hat{\sigma}_{u,i}^2 \right). \end{aligned} \quad (5.90)$$

Proposition 5.3. *The closed-form expression of the ML estimate of ζ_u is given by*

$$\hat{\zeta}_u = \frac{\sum_{i=1}^I \left((J_{u,i} - 1) \cdot \frac{\hat{\sigma}_{u,i}^2}{f(\hat{\mu}_{u,i}; a, b, \gamma)} \right)}{\sum_{i=1}^I (J_{u,i} - 1)}, \quad (5.91)$$

where $\hat{\mu}_{u,i}$ and $\hat{\sigma}_{u,i}^2$ are obtained from (5.77) and (5.87).

Proof. It is easy to derive from (5.89) and (5.90) that the ML estimate $\hat{\zeta}_u$ verifies

$$\frac{d\mathcal{L}}{d\zeta_u}(\hat{\zeta}_u) = -\frac{1}{2\hat{\zeta}_u} \cdot \sum_{i=1}^I \left((J_{u,i} - 1) - (J_{u,i} - 1) \cdot \frac{\hat{\sigma}_{u,i}^2}{f(\hat{\mu}_{u,i}; a, b, \gamma)} \cdot \frac{1}{\hat{\zeta}_u} \right) = 0. \quad (5.92)$$

We arrive at

$$\sum_{i=1}^I (J_{u,i} - 1) - \frac{1}{\hat{\zeta}_u} \cdot \sum_{i=1}^I \left((J_{u,i} - 1) \cdot \frac{\hat{\sigma}_{u,i}^2}{f(\hat{\mu}_{u,i}; a, b, \gamma)} \right) = 0, \quad (5.93)$$

hence the expression of $\hat{\zeta}_u$ as in (5.91). \square

Since $\hat{\sigma}_{u,i}^2$ is Gamma distributed as in (5.88), we can further derive from (5.91) the distribution of $\hat{\zeta}_u$ as

$$\hat{\zeta}_u \sim \mathcal{G} \left(\frac{1}{2} \cdot \sum_{i=1}^I (J_{u,i} - 1), \frac{2 \cdot \zeta_u}{\sum_{i=1}^I (J_{u,i} - 1)} \right), \quad (5.94)$$

where $\mathcal{G}(\eta, \lambda)$ denotes the Gamma distribution with pdf (5.65). As such, the expectation and variance of $\hat{\zeta}_u$ are expressed by

$$E[\hat{\zeta}_u] = \zeta_u \quad \text{and} \quad \text{var}[\hat{\zeta}_u] = \frac{2 \cdot \zeta_u^2}{\sum_{i=1}^I (J_{u,i} - 1)}. \quad (5.95)$$

These expressions imply that (5.91) is an unbiased estimator of ζ_u , and that the variance is very small because of the large enough number of considered IID pixels $\sum_{i=1}^I (J_{u,i} - 1)$ and $\zeta_u \leq 1$. In other words, the proposed ML estimate is of high accuracy, and can be treated as constant. Yet, since a , b and γ are still unknown, we cannot applied directly (5.91) to find $\hat{\zeta}_u$ in practice. This is why we propose to replace (a, b, γ) by $(\hat{a}, \hat{b}, \hat{\gamma})$, and eventually obtain the ML estimate of ζ_u as

$$\hat{\zeta}_u = \frac{\sum_{i=1}^I \left((J_{u,i} - 1) \cdot \frac{\hat{\sigma}_{u,i}^2}{f(\hat{\mu}_{u,i}; \hat{a}, \hat{b}, \hat{\gamma})} \right)}{\sum_{i=1}^I (J_{u,i} - 1)}, \quad (5.96)$$

where the triplet $(\hat{a}, \hat{b}, \hat{\gamma})$ is given from (5.84).

5.4.2 GLRT-Based Detector and Performance

Given the constant ML estimates $\hat{\boldsymbol{\mu}}_u = (\hat{\mu}_{u,1}, \dots, \hat{\mu}_{u,i}, \dots, \hat{\mu}_{u,I})$, $(\hat{a}, \hat{b}, \hat{\gamma})$ and $\hat{\zeta}_u$, we would like to propose in this subsection a more practical GLRT-based detector and to evaluate its statistical performance. Indeed, substituting $(\mu_{u,i}, a, b, \gamma, \zeta_u)$ by $(\hat{\mu}_{u,i}, \hat{a}, \hat{b}, \hat{\gamma}, \hat{\zeta}_u)$, the Corollary 5.1 leads to the following GLRT-based detector.

Corollary 5.2. *Under the practical context where all the model parameters are unknown except the magnification factor p , the optimal detector to distinguish an original TIFF image from a resampled TIFF image is as follows*

$$\hat{\delta}^*(\mathbf{r}_u) = \begin{cases} \mathcal{H}_0 : \text{original TIFF image} & \text{if } \hat{\Lambda}(\mathbf{r}_u) < \hat{\theta}_u, \\ \mathcal{H}_1 : \text{resampled TIFF image} & \text{if } \hat{\Lambda}(\mathbf{r}_u) \geq \hat{\theta}_u. \end{cases} \quad (5.97)$$

where $u \in \{2, \dots, p\}$, $\hat{\Lambda}(\mathbf{r}_u)$ denotes the generalized LLR of \mathbf{r}_u given by

$$\hat{\Lambda}(\mathbf{r}_u) = -\frac{1}{2} \cdot \left(\frac{1}{\hat{\zeta}_u} - 1 \right) \cdot \sum_{i=1}^I \left(\frac{1}{f(\hat{\mu}_{u,i}; \hat{a}, \hat{b}, \hat{\gamma})} \cdot \sum_{j=1}^{J_{u,i}} r_{u,i,j}^2 \right) + \rho_u \cdot \log \frac{1}{\hat{\zeta}_u}, \quad (5.98)$$

and $\hat{\theta}_u$ is computed as

$$\hat{\theta}_u = \bar{F}_{\hat{\Lambda}(\mathbf{r}_u), \mathcal{H}_0}^{-1}(\alpha_0), \quad (5.99)$$

with $\bar{F}_{\hat{\Lambda}(\mathbf{r}_u), \mathcal{H}_0}^{-1}(\cdot)$ the inverse function of $\bar{F}_{\hat{\Lambda}(\mathbf{r}_u), \mathcal{H}_0}(\cdot)$ defined by

$$\bar{F}_{\hat{\Lambda}(\mathbf{r}_u), \mathcal{H}_0}(\hat{\theta}_u) = G\left(\rho_u \cdot \log \frac{1}{\hat{\zeta}_u} - \hat{\theta}_u; \rho_u, \frac{1}{\hat{\zeta}_u} - 1\right), \quad (5.100)$$

in which $G(\cdot)$ is the Gamma cdf (5.58), and ρ_u is given from (5.59).

Similar to the LRT-based detector, the power function of the GLRT-based detector $\hat{\delta}^*(\mathbf{r}_u)$ is expressed by

$$\beta\left(\hat{\delta}^*(\mathbf{r}_u)\right) = G\left(\rho_u \cdot \log \frac{1}{\hat{\zeta}_u} - \hat{\theta}_u; \rho_u, 1 - \hat{\zeta}_u\right), \quad (5.101)$$

where $\hat{\theta}_u$ is computed following (5.99), $G(\cdot)$ is the Gamma cdf (5.58), and ρ_u is obtained by (5.59). The power (5.101) is a non-decreasing function of ρ_u and a non-increasing function of $\hat{\zeta}_u$.

5.5 Numerical Experiments

The aim of this section is twofold: (i) numerical validation and illustration of the proposed GLRT-based detector, and (ii) numerical assessment of the detector performance under various scenarios of interpolation kernels and sizes of resampled TIFF images.

5.5.1 Validation and Illustration

The validation and illustration rely on the Dresden images dataset [78]. Firstly, we convert the RAW images of the dataset to 8-bits TIFF color images using <https://image.online-convert.com/convert-to-tiff>. Such images are considered as original. Next, we resample them following the horizontal direction with resampling factor $\xi = \frac{p}{q}$ using some well-known interpolation methods (e.g., linear, cubic convolution, nearest-neighbor, spline, etc.), and save again in TIFF format to obtain the resampled TIFF images. Among the three color channels of TIFF images, we only consider the green one because the correlation among pixels generated by the demosaicing is smaller than in the red and blue channels. Consequently, the denoised pixels and residual noises in sub-images of the green channel are used as input of our experiments. As an illustration, an original TIFF image with size 3892×2608 issued by Nikon D200 camera is considered and horizontally resampled at rate $\xi = \frac{p}{q} = \frac{4}{3}$ using the cubic convolution interpolation [103] (see also Table 5.3 and Figure 5.9 for the expression and the shape of the cubic convolution kernel). Figure 5.6 shows four sub-images horizontally extracted from the green channel of the original and resampled TIFF images. These sub-images result in the illustrations in the sequel of this subsection.

We would like to verify the uniform distribution (5.43) of $t_{u,i,j}$ and the Gaussian distribution (5.46) of $r_{u,i,j}$. The process described in Subsection 5.3.2 with $I = 2^8 = 256$ and $\Delta_i = 1$ is first used to extract $t_{u,i,j}$ and $r_{u,i,j}$ from the above sub-images. Next, on the one hand, we apply the *kernel density estimation* (KDE) method [23] to find the empirical pdf of $t_{u,i,j}$ and $r_{u,i,j}$. On the other hand, we use the ML method to estimate the pdfs under the assumption that $t_{u,i,j}$ and $r_{u,i,j}$ are uniformly and Gaussian

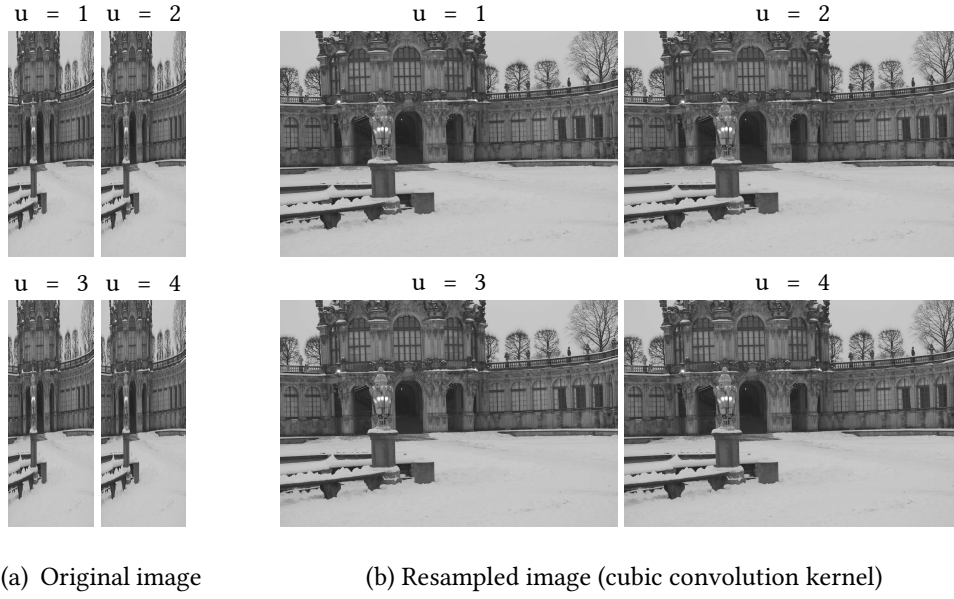


Figure 5.6: Sub-images Z_u , $u = 1, \dots, 4$, of green channel given from TIFF images issued by Nikon D200 camera

distributed respectively. Such a process is applied for all sub-images of both the original and resampled TIFF images. The comparison between the empirical and estimated pdfs of $t_{u,i,j}$ and $r_{u,i,j}$ allow us to confirm the relevance of conjecture. As an example, we consider the denoised pixels $t_{u,i,j}$ and the residues $r_{u,i,j}$ associated with the homogeneous segment $i = 80$ of the sub-images in Figure 5.6, and display in Figures 5.7a and 5.7b their pdfs. Clearly, the pdf curves returned by both the KDE and ML

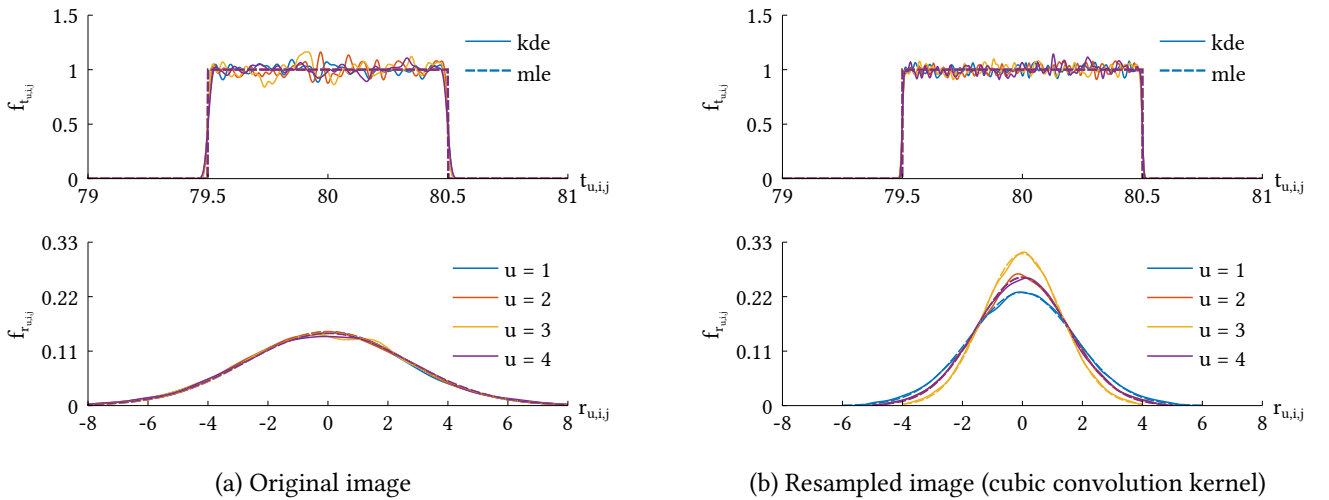


Figure 5.7: Probability density function of residues $r_{u,i,j}$ and denoised pixels $t_{u,i,j}$ in the segment $i = 80$

methods are almost identical. The figures on top show that $t_{u,i,j}$ are uniformly distributed between 79.5 and 80.5 with expectation $E[t_{u,i,j}] = 80$ and with very small variance $\text{var}[t_{u,i,j}] = \frac{1}{12}$, $\forall u \in \{1, \dots, 4\}$. This is why the sample mean $\hat{\mu}_{u,i}$ in (5.77) can be seen as the true expectation $\mu_{u,i}$. Observing the figures at the bottom, we find that $r_{u,i,j}$ are zero-mean Gaussian distributed with variance $\sigma_{u,i}^2$. The

variances are the same for all sub-images $\sigma_{1,i}^2 = \sigma_{2,i}^2 = \sigma_{3,i}^2 = \sigma_{4,i}^2$ when the TIFF image is original (see Figure 5.7a). Meanwhile, they are well distinct for the resampled TIFF image (see Figure 5.7b): the one of interpolated sub-images is actually smaller than the one of original sub-image $\sigma_{u,i}^2 < \sigma_{1,i}^2$, $\forall u = 2, 3, 4$. All the observations confirm the pertinence of our conjecture on $t_{u,i,j}$ and $r_{u,i,j}$.

Now, we deal with the relevance of the choice of sub-images for model parameters estimation and for the hypothesis testing. In fact, we have chosen the original sub-image \mathbf{Z}_1 as input of the estimation for image parameters $\boldsymbol{\mu}_1$ and camera parameters (a, b, γ) , and the interpolated sub-images $\mathbf{Z}_u, u \in \{2, \dots, p\}$ to estimate the image parameters $\boldsymbol{\mu}_u$, the sum of squared interpolation coefficients ζ_u and to serve the (G)LRT. The logic of this choice for a certain homogeneous segment can be justified by Figure 5.7. Indeed, the same pdf curves of $r_{u,i,j}$ in Figure 5.7a imply that the camera parameters (a, b, γ) are unchanged for all sub-images of a given TIFF image. So, a simplest way to find their estimates $(\hat{a}, \hat{b}, \hat{\gamma})$ is to use \mathbf{Z}_1 as it is independent of interpolation coefficients. Once obtained, we can use them as parameters of models of interpolated sub-images $\mathbf{Z}_u, u \in \{2, 3, 4\}$. Similarly, the distinction among the pdf curves of $r_{u,i,j}$ when $u = 1$ and $u \neq 1$ in Figure 5.7b confirms why $\mathbf{Z}_u, u \in \{2, \dots, p\}$, are served to estimate ζ_u and to design the (G)LRT-based detector. Especially, we find that the pdf curves for $u = 2$ and $u = 4$ almost coincide because they share the same value of $\zeta_u = 0.8088$ (see also Table 5.1). The above logic remains true when we extend to the overall sub-images (i.e., for all homogeneous segments). In fact, looking at the scatter-plot of $\left\{ \hat{\mu}_{u,i}, \hat{\sigma}_{u,i}^2 \right\}_{i=1}^I$ for each sub-images $\mathbf{Z}_u, u \in \{1, \dots, 4\}$ of both the considered original and resampled TIFF images in Figure 5.6, we exactly find the same phenomena as above. This strengthens the relevance in the choice of sub-images.

To show the efficiency of the proposed ML estimators (5.84), we evaluate how good the fitted curves $\sigma_{u,i}^2 = f(\mu_{u,i}; \zeta_u \cdot a, \zeta_u \cdot b, \gamma)$, with $\zeta_u \leq 1$, of the scatter-plot of $\left\{ \hat{\mu}_{u,i}, \hat{\sigma}_{u,i}^2 \right\}_{i=1}^I$ are. As displayed in Figure 5.8, we can always find a function $f(\cdot)$ that fits in very well with the scatter-plot of a sub-image. This implies the accuracy of the estimators (5.84). The estimates $(\hat{a}, \hat{b}, \hat{\gamma})$ given from \mathbf{Z}_1 of

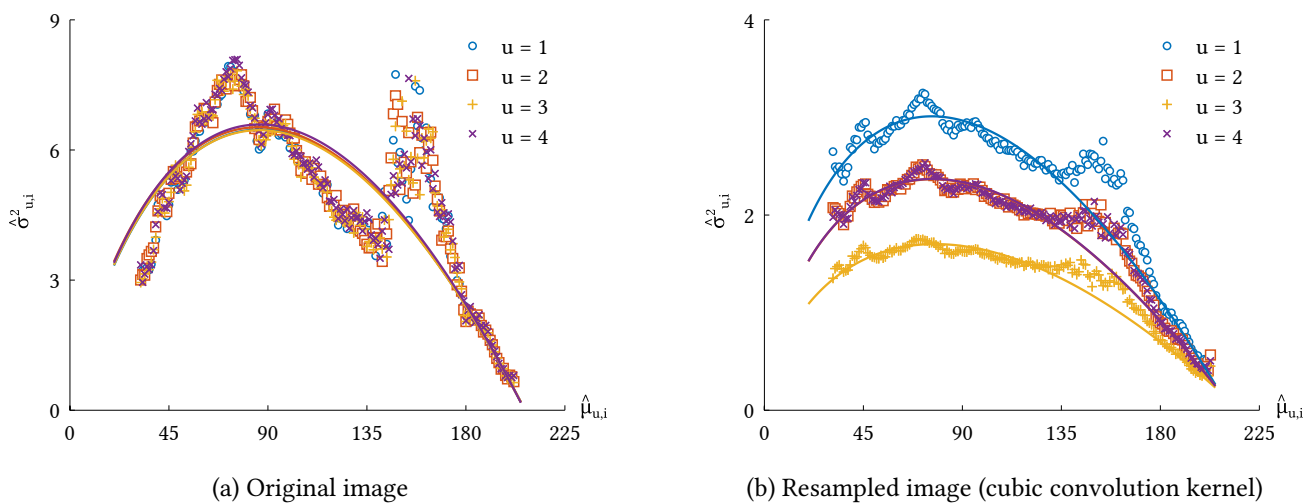


Figure 5.8: Scatter-plot and fitted curve of the variances and expectations of pixels given from sub-images \mathbf{Z}_u

the original and resampled TIFF images are provided in Table 5.1. We note that their values are not

TIFF image	sub-image	\hat{a}	\hat{b}	\hat{y}	$\hat{\zeta}_u$	ζ_u
original	1	-0.0051	0.0991	0.5559	-	-
	2	-	-	-	1	1
	3	-	-	-	0.9985	1
	4	-	-	-	1	1
resampled	1	-0.0040	0.1371	0.6578	-	-
	2	-	-	-	0.8186	0.8088
	3	-	-	-	0.6343	0.6406
	4	-	-	-	0.8191	0.8088

Table 5.1: Parameters estimated from sub-images \mathbf{Z}_u , $u \in \{1, \dots, 4\}$, of original and resampled TIFF images

necessary identical because the content of the associated sub-images are not the same. In the same table, we find $\hat{\zeta}_u$ estimated from \mathbf{Z}_u , $u \in \{2, 3, 4\}$. Compared to the true value ζ_u , the estimated ones are almost similar. Moreover, $\hat{\zeta}_u$ is near to 1 when \mathbf{Z}_u are original, and is definitely less than 1 when \mathbf{Z}_u are interpolated. All these elements confirm the accuracy of (5.96).

	image	original			resampled		
	sub-image	$u = 2$	$u = 3$	$u = 4$	$u = 2$	$u = 3$	$u = 4$
α_0	$\hat{\Lambda}(\mathbf{r}_u)$	-0.0776	-5.9999	-0.1578	$2.6756 \cdot 10^3$	$1.1575 \cdot 10^4$	$2.8096 \cdot 10^3$
0.05	$\hat{\theta}_u$	0.0156	1.3371	0.0157	$-2.9883 \cdot 10^3$	$-1.3125 \cdot 10^4$	$-3.1375 \cdot 10^3$
	$\hat{\delta}^*(\mathbf{r}_u)$	\mathcal{H}_0	\mathcal{H}_0	\mathcal{H}_0	\mathcal{H}_1	\mathcal{H}_1	\mathcal{H}_1
0.5	$\hat{\theta}_u$	$-4.2150 \cdot 10^{-5}$	-1.0739	$-4.2279 \cdot 10^{-5}$	$-3.1213 \cdot 10^3$	$-1.3407 \cdot 10^4$	$-3.2738 \cdot 10^3$
	$\hat{\delta}^*(\mathbf{r}_u)$	\mathcal{H}_0	\mathcal{H}_0	\mathcal{H}_0	\mathcal{H}_1	\mathcal{H}_1	\mathcal{H}_1
0.95	$\hat{\theta}_u$	-0.0157	-3.4876	-0.0158	$-3.2544 \cdot 10^3$	$-1.3690 \cdot 10^4$	$-3.4102 \cdot 10^3$
	$\hat{\delta}^*(\mathbf{r}_u)$	\mathcal{H}_0	\mathcal{H}_0	\mathcal{H}_0	\mathcal{H}_1	\mathcal{H}_1	\mathcal{H}_1

Table 5.2: GLRT using sub-images \mathbf{Z}_u , $u \in \{2, 3, 4\}$

Table 5.2 shows the values of the LLR $\hat{\Lambda}(\mathbf{r}_u)$ and of the decision threshold $\hat{\theta}_u$, as well as the results returned by the GLRT-based detector $\hat{\delta}^*(\mathbf{r}_u)$ when the false-alarm rate α_0 is 0.05, 0.5 and 0.95 respectively. With a fixed value of $\hat{\Lambda}(\mathbf{r}_u)$, the decrease of $\hat{\theta}_u$ is in tune with the increase of α_0 . We find that the detector returns the right results for whatever used sub-images \mathbf{Z}_u , $u \in \{2, 3, 4\}$. However, considering the difference between $\hat{\Lambda}(\mathbf{r}_u)$ and $\hat{\theta}_u$, the sub-images \mathbf{Z}_3 gives the most robust result. This actually makes sense as \mathbf{Z}_3 has the smallest $\hat{\zeta}_u$ compared to the two others (see Table 5.1). In other words, we should carry out the GLRT on the sub-image which has the smallest sum of square interpolation coefficients to ensure the highest detection performance.

5.5.2 Performance Assessment

From (5.101), the power function $\beta(\hat{\delta}^*(\mathbf{r}_u))$ only depends on $\hat{\zeta}_u$ and ρ_u . As such, the interpolation methods and the size of resampled TIFF images are the main factors that influence the statistical

performance of the GLRT-based detector, and we would like to understand their impacts through sensitivity studies. We consider therefore 1000 authentic 8-bits color TIFF images of size varied from 1920×1080 pixels to 3888×2592 pixels. They are gathered from several well-known databases such as the McGill Calibrated Colour Image Database [162], the Dresden Image Database [78], the Kodak Lossless True Color Image Suite⁸, the Image Manipulation Dataset [41], the DSO-1 and DSI-1 Datasets [52], and the Realistic Tampering Dataset [115]. These original images are squarely cropped in several sizes, next resampled at rate $\xi = \frac{p}{q} = \frac{4}{3}$ using some well-known interpolation methods, and are finally saved again in TIFF format to create tested datasets of resampled TIFF images.

name	expression	support
nearest neighbor	$h(\delta) = \begin{cases} 1 & \text{if } \delta \leq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$	$[-\frac{1}{2}, \frac{1}{2}]$
linear	$h(\delta) = \begin{cases} 1 - \delta & \text{if } \delta \leq 1 \\ 0 & \text{otherwise} \end{cases}$	$[-1, 1]$
cubic convolution	$h(\delta) = \begin{cases} \frac{3}{2} \delta ^3 - \frac{5}{2} \delta ^2 + 1 & \text{if } \delta \leq 1 \\ -\frac{1}{2} \delta ^3 + \frac{5}{2} \delta ^2 - 4 \delta + 2 & \text{if } 1 < \delta \leq 2 \\ 0 & \text{otherwise} \end{cases}$	$[-2, 2]$

Table 5.3: Well-known interpolation kernels

Four interpolation methods (i.e., nearest neighbor, linear, cubic convolution and cubic spline) are under consideration [145, 164]. The three first methods can be simply defined by an associated inter-

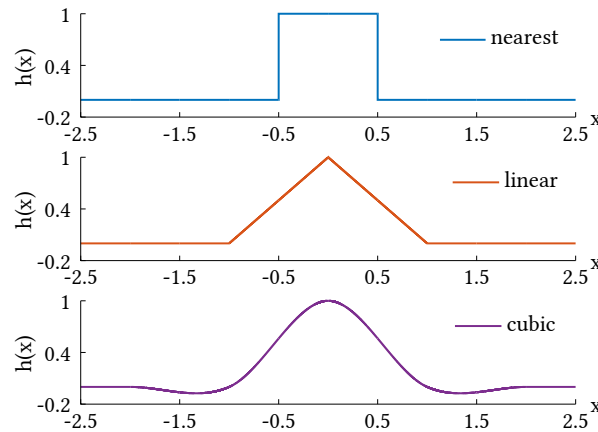


Figure 5.9: Shape of well-known interpolation kernels

polation kernel (see Table 5.3 and Figure 5.9), meanwhile the definition of the last one is more sophisticated and can be found in [51]. The size of all tested resampled TIFF images is fixed at 1024×1024 . Figures 5.10a, 5.10b and 5.10c represent the performance of the GLRT-based detector in function of the false-alarm rate α_0 when the three sub-images Z_u , $u \in \{2, 3, 4\}$, are respectively used as input. In each

⁸https://www.math.purdue.edu/~lucier/PHOTO_CD/TIFF_IMAGES/

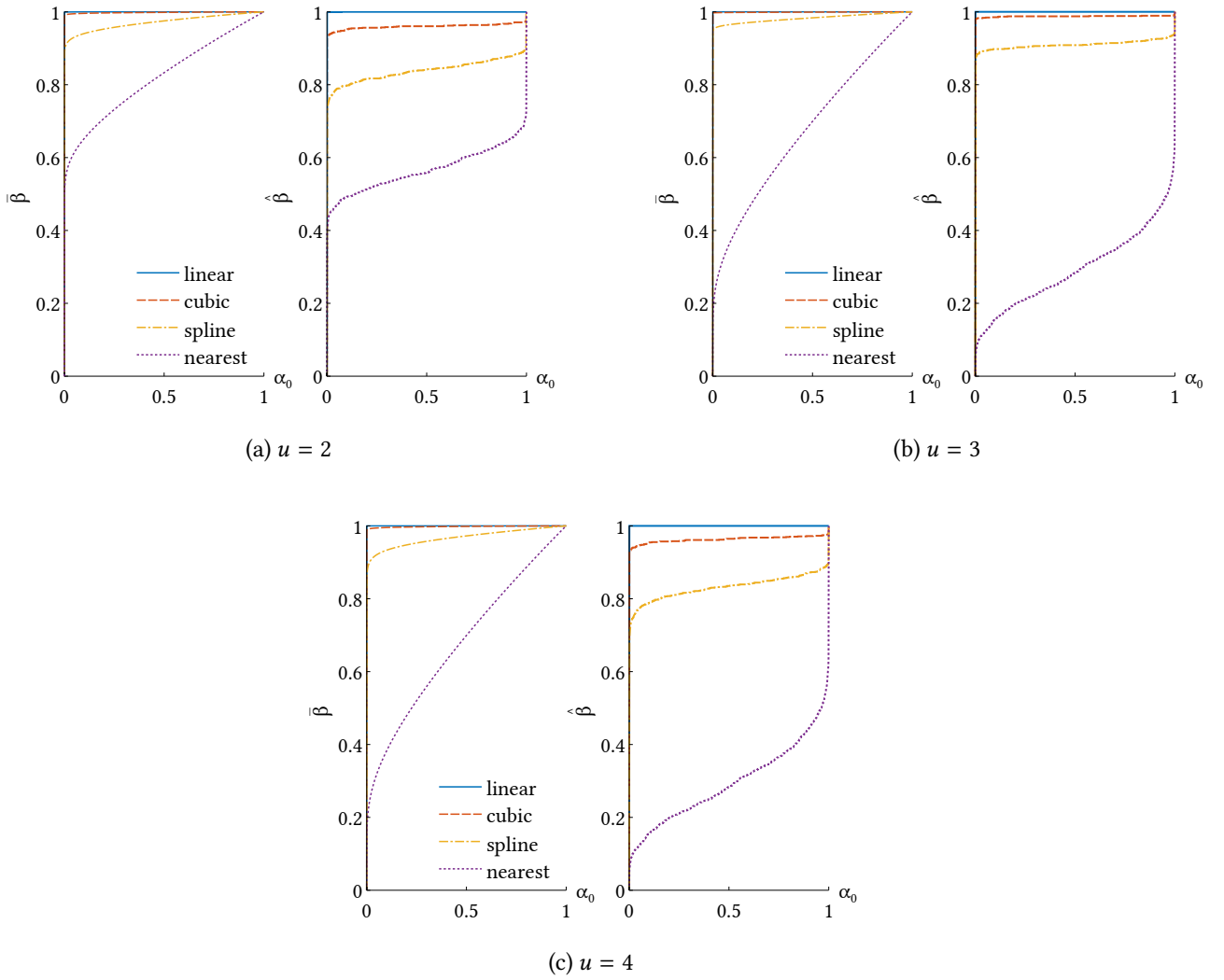


Figure 5.10: Power functions given from sub-images \mathbf{Z}_u , $u \in \{2, 3, 4\}$ with respect to interpolation methods

figure, the sub-figure on the right displays the empirical power function $\hat{\beta} = \frac{1}{1000} \sum_{m=1}^{1000} \mathbf{1}_{\{\hat{\delta}_m^*(\mathbf{r}_u) = \mathcal{H}_1\}}$ where $\hat{\delta}_m^*(\mathbf{r}_u)$ denotes the decision on the m -th image returned by the rule (5.97), while the sub-figure on the left shows the power average over 1000 images $\bar{\beta} = \frac{1}{1000} \sum_{m=1}^{1000} \beta(\hat{\delta}_m^*(\mathbf{r}_u))$ with $\beta(\hat{\delta}_m^*(\mathbf{r}_u))$ computed by (5.101). Both $\hat{\beta}$ and $\bar{\beta}$ increase from 0 to 1 with respect to α_0 , and $\hat{\beta} \leq \bar{\beta}$ for a given interpolation method. Comparing $\hat{\beta}$ and $\bar{\beta}$ associated with different considered interpolation methods for all sub-images \mathbf{Z}_u , $u \in \{2, 3, 4\}$, we find that the stronger the “smoothing” effect of interpolation, the lower the power of the GLRT-based detector. This implies that the TIFF images resampled using linear interpolation is easier to be detected by the proposed detector than the ones using cubic convolution or cubic spline interpolation. The nearest neighbor interpolation is however a special case where the resampling is hard to be detected even if the “smoothing” effect is weak. All these phenomena are comprehensible when we observe the sample mean $E[\hat{\zeta}_u]$ and sample variance $\text{var}[\hat{\zeta}_u]$ of $\hat{\zeta}_u$ in Table 5.4. As the values of $\text{var}[\hat{\zeta}_u]$ are very small, $E[\hat{\zeta}_u]$ approximates well $\hat{\zeta}_u$. We can therefore apply the

sub-image	method	linear	cubic convolution	cubic spline	nearest neighbor
$u = 2$	$E[\hat{\zeta}_u]$	0.8216	0.9434	0.9676	0.9844
	$\text{var}[\hat{\zeta}_u]$	$6.0547 \cdot 10^{-4}$	$3.5575 \cdot 10^{-4}$	$3.2439 \cdot 10^{-4}$	$3.0216 \cdot 10^{-4}$
$u = 3$	$E[\hat{\zeta}_u]$	0.7674	0.9075	0.9513	0.9959
	$\text{var}[\hat{\zeta}_u]$	$1.0687 \cdot 10^{-3}$	$7.5549 \cdot 10^{-4}$	$5.5014 \cdot 10^{-4}$	$5.1991 \cdot 10^{-5}$
$u = 4$	$E[\hat{\zeta}_u]$	0.8270	0.9508	0.9748	0.9959
	$\text{var}[\hat{\zeta}_u]$	$6.5531 \cdot 10^{-4}$	$2.9480 \cdot 10^{-4}$	$2.4549 \cdot 10^{-4}$	$5.1991 \cdot 10^{-5}$

Table 5.4: Sample mean $E[\hat{\zeta}_u]$ and sample variance $\text{var}[\hat{\zeta}_u]$ of $\hat{\zeta}_u$ over 1000 tested resampled TIFF images

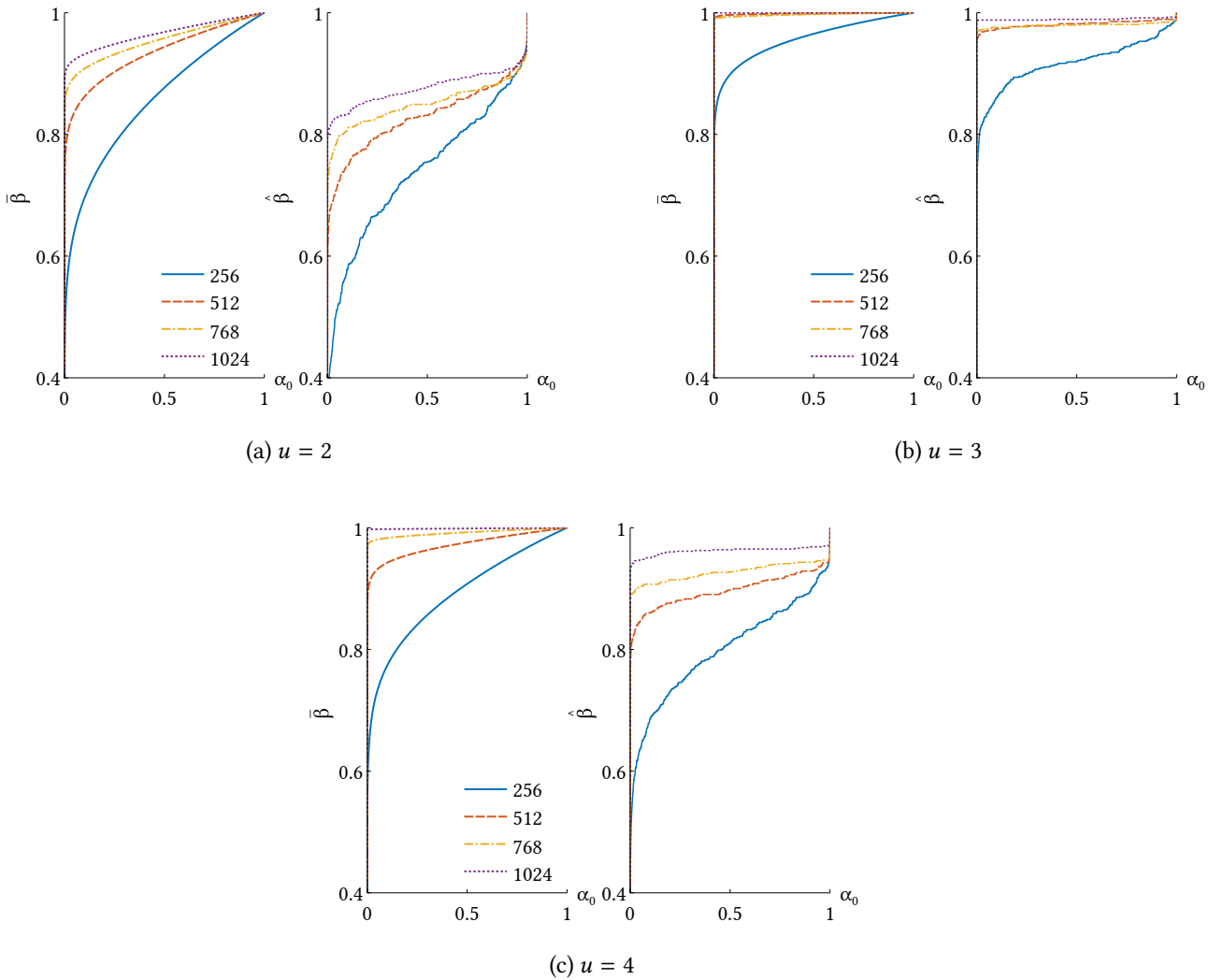


Figure 5.11: Power functions given from sub-images Z_u , $u \in \{2, 3, 4\}$ with respect to TIFF image sizes

Proposition 5.2 to $E[\hat{\zeta}_u]$ and the above results are consequent.

Now, fixing the interpolation method for resampled TIFF images at cubic convolution, and varying their sizes as 256×256 , 512×512 , 768×768 and 1024×1024 , we sketch in Figures 5.11a, 5.11b and 5.11c the power functions $\hat{\beta}$ and $\bar{\beta}$ computed from the sub-images Z_u , $u \in \{2, 3, 4\}$, respectively. The meanings of $\hat{\beta}$ and $\bar{\beta}$ are similar to the ones in Figures 5.10a, 5.10b and 5.10c. We can find clearly

that the statistical performance of the GLRT-based detector increases with respect to the size of tested images. This is absolutely coherent with Proposition 5.2 as the total number of residual noises under consideration ρ_u increases when the image size increases.

5.6 Conclusion

By analyzing the processing process from a RAW image to a resampled TIFF image, we develop in this chapter statistical models for the noisy pixels, the denoised pixels, as well as for the residual noises of resampled TIFF images. An estimation process is proposed to specify the parameters of these models. Based on the model of residual noises, we design optimal detectors to distinguish a resampled TIFF image from an original TIFF image using the incoherence in their variance. Consequently, a LRT-based detector and a GLRT-based detector are proposed for the ideal and practical context where the model parameters are respectively known and unknown. The closed-form expressions of their power function are also provided and analyzed. Various numerical experiments show that the detectors performance depends closely on the size and the order of sub-images under the consideration. So, we should take care about them to reach the highest performance of the proposed detectors.

Chapter 6

Conclusions and Perspectives

6.1 Conclusions

We have developed in this doctoral project three versatile techniques for detecting forged digital images and localizing various types of tampering therein: (i) *demosaicing traces-based* [126], (ii) *JPEG-compression traces-based* [127, 128], and (iii) *resampling traces-based* [125]. Although these techniques are different, they work under the common assumption that manipulations may alter some underlying statistical properties of natural images. The same two-steps process has been adopted for every detection technique: (i) analyze and model statistical features of both the authentic and forged images associated with specific in-camera and/or post-camera traces, then (ii) design a statistical detector to differentiate between the authentic and forged images by estimating statistical changes in their models.

Chapter 3 exploited the inconsistency in traces left by the demosaicing. The associated feature statistic is the standardized difference between the arithmetic mean of the logarithm of residues variances of acquired and interpolated pixels in square blocks within smoothing regions of a suspect digital image. Such a feature statistic follows the standard normal distribution and a mixture of two normal distributions when the image is authentic and forged respectively. Normality tests are next carried out to decide if the suspect image is authentic or not. If it is forged, a penalized EM algorithm is used to automatically distinguish between original and tampered area within the image.

When JPEG compression traces are considered as in Chapter 4, we defined the value of DCT coefficients as their feature statistic and developed its recurrent modeling framework for both the authentic and forged JPEG images subject to multiple quantization effect. Especially, we have proved that the quantized DCT coefficients of forged JPEG images can be properly approximated by a two-components mixture model regardless of the number of forged areas therein. Based on such a model, the plug-in Bayes rule combined with a simple EM algorithm is exploited to localize tampered areas of the forged image.

Finally, in Chapter 5, the noise part of pixels is chosen as the feature statistic of resampling traces. We have followed the complete processing process from a RAW image to a resampled TIFF image to find out that the noise models of both the resampled and original TIFF images are zero-mean normal

distributions. For independent identically distributed pixels, the variance of resampled noise is smaller proportionally to the sum of squared interpolation coefficients. These models allows us to design (G)LRT-based detectors for distinguishing between a resampled and an original TIFF images. Their performances are analytically evaluated.

We have carried out various numerical experiments on several well-known benchmark datasets to assess both the performances and robustness of the proposed detection techniques. We believe that this doctoral thesis provides a significant complement to the field of statistical image modeling and digital image forgery detection.

6.2 Perspectives

The thesis highlights three potential interests for future researches according to the three developed detection techniques.

1. Even if the demosaicing traces-based technique developed in Chapter 3 yields very encouraging results, we find that the automatic authentication of digital images is still at low performance (see Table 3.2), and that the forgery localization is limited to uncompressed or less-compressed images (see Figures 3.8 and 3.9). The focus of our near future work is to further improve these two points. Building a new feature statistic taking into account the periodicity of demosaicing artifacts in DCT domain as in [134] seems to be a key step for this work. Another perspective is to extend the developed methods to the videos forensics. The work of Singh and Aggarwal in [195] could be a good orientation.
2. Despite very encouraging results returned by the modeling framework of JPEG DCT coefficients developed in Chapter 4, several improvements can be made. Firstly, the truncation errors should not be ignored from the statistical modeling of JPEG compression cycle even its small probability. Some interesting ideas for the consideration of truncation errors can be found in [216]. Secondly, as in [221], the correlation among adjacent DCT blocks can be taken into account in modeling the distribution of AC DCT coefficients. For further perspectives, we can think about other consequences of manipulations on JPEG images (e.g., non-aligned recompression [18]). We can also extend the modeling approach used in Chapter 4 to model tampered MPEG videos [218].
3. Chapter 5 has focused on one-dimensional resampling. To deal with more practical digital image forgery, we should extend the proposed detection technique to two-dimensional resampling [180, 181]. Besides, the same technique could be applied to regions of images to detect the local resampling [169, 175].

Appendix A

Appendix of Chapter 3

This appendix provides the detailed formalization of the penalized EM algorithm used in Chapter 3. Indeed, given the parameters $\Theta^{(t)} = (\gamma^{(t)}, \mu_1^{(t)}, \mu_2^{(t)}, \sigma^{(t)})$ at the t -th iteration, the EM algorithm at the $(t + 1)$ -st iteration behaves as follows.

E-step of the EM algorithm

Using Bayes theorem, the posterior distribution of the latent variables is expressed by

$$f(\mathbf{Z} | \mathbf{N}) = \frac{f(\mathbf{N}, \mathbf{Z})}{f(\mathbf{N})} \propto \prod_{i=1}^q \prod_{k=1}^2 \left(\gamma_k^{(t)} f(N_i; \mu_k^{(t)}, \sigma^{(t)}) \right)^{Z_{ik}}. \quad (\text{A.1})$$

Taking the expectation of Z_{ik} under $f(\mathbf{Z} | \mathbf{N})$, we obtain [21, page 443]

$$E \left[Z_{ik} | \mathbf{N}; \Theta^{(t)} \right] = \frac{\gamma_k^{(t)} f(N_i; \mu_k^{(t)}, \sigma^{(t)})}{\sum_{k=1}^2 \gamma_k^{(t)} f(N_i; \mu_k^{(t)}, \sigma^{(t)})} \triangleq \gamma_{ik}^{(t+1)}. \quad (\text{A.2})$$

γ_{ik} represents the responsibility of the population k for the data point N_i . This gives an idea for population clustering. The expectation of $pl_c(\gamma, \mu_1, \mu_2, \sigma)$ under the posterior distribution of the latent variables is thus

$$E_{\mathbf{Z}} \left[pl_c(\gamma, \mu_1, \mu_2, \sigma) | \mathbf{N}; \Theta^{(t)} \right] = \sum_{i=1}^q \sum_{k=1}^2 E \left[Z_{ik} | \mathbf{N}; \Theta^{(t)} \right] \cdot \left(\log \gamma_k - \log \sigma - \frac{1}{2\sigma^2} (N_i - \mu_k)^2 - \frac{1}{2} \log 2\pi \right) + p_q(\sigma) = \sum_{i=1}^q \sum_{k=1}^2 \gamma_{ik}^{(t+1)} \cdot \left(\log \gamma_k - \log \sigma - \frac{1}{2\sigma^2} (N_i - \mu_k)^2 - \frac{1}{2} \log 2\pi \right) + p_q(\sigma) \triangleq Q(\Theta; \Theta^{(t)}). \quad (\text{A.3})$$

M-step of the EM algorithm

We look for $\Theta^{(t+1)}$ that maximizes $Q(\Theta; \Theta^{(t)})$. Since that γ, μ_1, μ_2 and σ being in separate linear terms, they may all be maximized independently.

- Considering only the terms of $Q(\Theta; \Theta^{(t)})$ in associated with γ , we have

$$g(\gamma) = \sum_{i=1}^q \left(\gamma_{i1}^{(t+1)} \log \gamma_1 + \gamma_{i2}^{(t+1)} \log \gamma_2 \right) = \log(1-\gamma) \sum_{i=1}^q \gamma_{i1}^{(t+1)} + \log \gamma \sum_{i=1}^q \gamma_{i2}^{(t+1)}. \quad (\text{A.4})$$

Taking the derivation of $g(\gamma)$ according to γ and setting $\frac{dh(\gamma)}{d\gamma} = 0$ lead to

$$\frac{dh(\gamma)}{d\gamma} = \frac{\sum_{i=1}^q \gamma_{i1}^{(t+1)}}{\gamma - 1} + \frac{\sum_{i=1}^q \gamma_{i2}^{(t+1)}}{\gamma} = 0 \Leftrightarrow \gamma = \frac{\sum_{i=1}^q \gamma_{i2}^{(t+1)}}{q}. \quad (\text{A.5})$$

Consequently, $\gamma^{(t+1)}$ can be updated by

$$\gamma^{(t+1)} = \frac{\sum_{i=1}^q \gamma_{i2}^{(t+1)}}{q}. \quad (\text{A.6})$$

- Considering only the terms of $Q(\Theta; \Theta^{(t)})$ in associated with μ_k , $k \in \{1, 2\}$, we have

$$s(\mu_k) = -\frac{1}{2\sigma^2} \sum_{i=1}^q \gamma_{ik}^{(t+1)} (N_i - \mu_k)^2. \quad (\text{A.7})$$

Taking the derivation of $s(\mu_k)$ according to μ_k and setting $\frac{ds(\mu_k)}{d\mu_k} = 0$ lead to

$$\frac{ds(\mu_k)}{d\mu_k} = \frac{\sum_{i=1}^q \gamma_{ik}^{(t+1)} (N_i - \mu_k)}{\sigma_k^2} = 0 \Leftrightarrow \sum_{i=1}^q \gamma_{ik}^{(t+1)} N_i = \mu_k \sum_{i=1}^q \gamma_{ik}^{(t+1)}. \quad (\text{A.8})$$

Setting $\frac{ds(\mu_k)}{d\mu_k} = 0$ leads to

$$\sum_{i=1}^q \gamma_{ik}^{(t+1)} N_i = \mu_k \sum_{i=1}^q \gamma_{ik}^{(t+1)}. \quad (\text{A.9})$$

Consequently, $\mu_k^{(t+1)}$, $k \in \{1, 2\}$, can be updated by

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^q \gamma_{ik}^{(t+1)} N_i}{\sum_{i=1}^q \gamma_{ik}^{(t+1)}}. \quad (\text{A.10})$$

- Setting $\mu_k = \mu_k^{(t+1)}$, and considering only the terms of $Q(\Theta; \Theta^{(t)})$ in associated with σ , we have

$$r(\sigma) = -q \log \sigma - \frac{1}{2\sigma^2} \sum_{i=1}^q \sum_{k=1}^2 \gamma_{ik}^{(t+1)} \left(N_i - \mu_k^{(t+1)} \right)^2 - a_q \cdot \left(\frac{s_q^2}{\sigma^2} + \log \left(\frac{\sigma^2}{s_q^2} \right) \right). \quad (\text{A.11})$$

Denoting $S^{(t+1)} = \sum_{i=1}^q \sum_{k=1}^2 \gamma_{ik}^{(t+1)} \left(N_i - \mu_k^{(t+1)} \right)^2$ and taking the derivation of $r(\sigma)$ according to

σ , we have

$$\frac{dr(\sigma)}{d\sigma} = -\frac{q + 2a_q}{\sigma} + \frac{S^{(t+1)} + 2a_q s_q^2}{\sigma^3}. \quad (\text{A.12})$$

Setting $\frac{dr(\sigma)}{d\sigma} = 0$ leads to

$$\frac{S^{(t+1)} + 2a_q s_q^2}{\sigma^2} = q + 2a_q. \quad (\text{A.13})$$

Consequently, $\sigma^{(t+1)}$ can be updated by

$$\sigma^{(t+1)} = \sqrt{\frac{S^{(t+1)} + 2a_q s_q^2}{q + 2a_q}}. \quad (\text{A.14})$$

The above developments give the key steps of the Algorithm 3.1.

Appendix B

Appendix of Chapter 4

This appendix recalls the statistical law and parameters estimation of the models for primary unquantized AC DCT coefficients used in Chapter 4. Although four models are considered, we just mention here the GGD and CND, because the LD and GND can be derived from GGD as GND $(\beta, \gamma) \equiv \text{GGD}\left(\frac{1}{\gamma}, \beta, \gamma\right)$ and LD $(\beta) \equiv \text{GGD}(1, \beta, 1)$.

Generalized Gamma Distribution

Let consider $X^{(1)} \sim \text{GGD}(\alpha, \beta, \gamma)$ with $(\alpha, \beta, \gamma) \in \mathbb{R}_{*,+}^3$, then the pdf and cdf of $X^{(1)}$ at $x^{(1)} \in \mathbb{R}$ are respectively

$$f_{X^{(1)}}\left(x^{(1)}; \alpha, \beta, \gamma\right) = \frac{\gamma}{2\beta\Gamma(\alpha)} \left(\frac{|x^{(1)}|}{\beta}\right)^{\gamma\alpha-1} e^{-\left(\frac{|x^{(1)}|}{\beta}\right)^\gamma}, \quad (\text{B.1})$$

$$F_{X^{(1)}}\left(x^{(1)}; \alpha, \beta, \gamma\right) = \frac{1}{2} \frac{\Gamma\left(\alpha, \left(\frac{-x^{(1)}}{\beta}\right)^\gamma\right)}{\Gamma(\alpha)} \cdot 1_{\{x^{(1)} < 0\}} + \frac{1}{2} \cdot 1_{\{x^{(1)} = 0\}} + \left(1 - \frac{1}{2} \frac{\Gamma\left(\alpha, \left(\frac{x^{(1)}}{\beta}\right)^\gamma\right)}{\Gamma(\alpha)}\right) \cdot 1_{\{x^{(1)} > 0\}}, \quad (\text{B.2})$$

where $1_{\{\cdot\}}$ denotes the indicator function which equals 1 if the argument is true and 0 otherwise, $\Gamma(u) = \int_0^\infty t^{u-1} e^{-t} dt$ and $\Gamma(u, x) = \int_x^\infty t^{u-1} e^{-t} dt$ are respectively the *complete* and *upper incomplete* gamma functions. Let $\widehat{\mathbf{x}}^{(1)} = \{\widehat{x}_1^{(1)}, \dots, \widehat{x}_N^{(1)}\}$ be the set of unquantized AC DCT coefficients recovered from a JPEG image (see Figure 4.2), the ML estimate of γ is the solution of the equation

$$\psi\left(\frac{\frac{1}{N} \sum_{n=1}^N |\widehat{x}_n^{(1)}|^\gamma}{\vartheta(\gamma)}\right) + \ln(\vartheta(\gamma)) - \frac{1}{N} \sum_{n=1}^N \ln |\widehat{x}_n^{(1)}|^\gamma = 0, \quad (\text{B.3})$$

where $\psi(\alpha) = \frac{\Gamma'(\alpha)}{\Gamma(\alpha)}$ denotes the digamma function and $\vartheta(\gamma)$ is defined by

$$\vartheta(\gamma) = \gamma \cdot \left(\frac{1}{N} \sum_{n=1}^N |\widehat{x}_n^{(1)}|^\gamma \ln |\widehat{x}_n^{(1)}| - \frac{1}{N} \sum_{n=1}^N \ln |\widehat{x}_n^{(1)}| \cdot \frac{1}{N} \sum_{n=1}^N |\widehat{x}_n^{(1)}|^\gamma\right). \quad (\text{B.4})$$

Once $\widehat{\gamma}$ is numerically obtained, the ML estimates of α and β are obtained by

$$\widehat{\alpha} = \frac{\frac{1}{N} \sum_{n=1}^N \left| \widehat{x}_n^{(1)} \right|^{\widehat{\gamma}}}{\vartheta(\widehat{\gamma})} \quad \text{and} \quad \widehat{\beta} = \left(\frac{1}{\widehat{\alpha}} \cdot \frac{1}{N} \sum_{n=1}^N \left| \widehat{x}_n^{(1)} \right|^{\widehat{\gamma}} \right)^{\frac{1}{\widehat{\gamma}}}. \quad (\text{B.5})$$

Compound Normal Distribution

Let $X^{(1)} \sim \text{CND}(\eta, \rho)$ with $(\eta, \rho) \in \mathbb{R}_{*,+}^2$, then the pdf of $X^{(1)}$ at $x^{(1)} \in \mathbb{R}$ is

$$f_{X^{(1)}}(x^{(1)}; \eta, \rho) = \sqrt{\frac{2}{\pi}} \frac{\left(|x^{(1)}| \sqrt{\frac{\rho}{2}} \right)^{\eta - \frac{1}{2}}}{\rho^\eta \Gamma(\eta)} K_{\eta - \frac{1}{2}} \left(|x^{(1)}| \sqrt{\frac{2}{\rho}} \right), \quad (\text{B.6})$$

where $K_\mu(\cdot)$ denotes the modified Bessel function of the second kind with order $\mu \geq -\frac{1}{2}$ [117, Appendix, page 315]

$$K_\mu(u) = \frac{\sqrt{\pi} u^\mu}{2^\mu \Gamma(\mu + \frac{1}{2})} \int_1^\infty (t^2 - 1)^{\mu - \frac{1}{2}} e^{-ut} dt. \quad (\text{B.7})$$

The cdf of $X^{(1)}$ at $x^{(1)} \in \mathbb{R}$ is

$$F_{X^{(1)}}(x^{(1)}; \eta, \rho) = \frac{1}{2} \left(1 - \mathcal{G}(-x^{(1)}) \right) \cdot \mathbf{1}_{\{x^{(1)} < 0\}} + \frac{1}{2} \cdot \mathbf{1}_{\{x^{(1)} = 0\}} + \frac{1}{2} \left(1 + \mathcal{G}(x^{(1)}) \right) \cdot \mathbf{1}_{\{x^{(1)} > 0\}}, \quad (\text{B.8})$$

where $\mathcal{G}(x)$ is defined by

$$\mathcal{G}(x) = g(x) \cdot \left(K_{\eta - \frac{1}{2}}(g(x)) \cdot \mathbf{L}_{\eta - \frac{3}{2}}(g(x)) + \mathbf{L}_{\eta - \frac{1}{2}}(g(x)) \cdot K_{\eta - \frac{3}{2}}(g(x)) \right), \quad (\text{B.9})$$

with $g(x) = x \sqrt{\frac{2}{\rho}}$ and $\mathbf{L}_\mu(\cdot)$ the modified Struve function with order μ [81, 8.550.2, page 942]

$$\mathbf{L}_\mu(u) = \sum_{k=1}^{\infty} \frac{\left(\frac{u}{2}\right)^{2k+\mu+1}}{\Gamma(k + \frac{3}{2}) \Gamma(\mu + k + \frac{3}{2})}. \quad (\text{B.10})$$

The ML estimates $\widehat{\eta}$ and $\widehat{\rho}$ are the simultaneous solutions of the following equations

$$2\eta - \frac{1}{N} \sum_{n=1}^N g\left(\left|\widehat{x}_n^{(1)}\right|\right) R_{\eta - \frac{1}{2}}\left(g\left(\left|\widehat{x}_n^{(1)}\right|\right)\right) = 0, \quad (\text{B.11})$$

and

$$\begin{aligned} \psi(\eta) + \ln 2 - \frac{1}{N} \sum_{n=1}^N \ln\left(g\left(\left|\widehat{x}_n^{(1)}\right|\right)\right) - \frac{1}{2} \left(\eta - \frac{1}{2} \right) \frac{1}{N} \sum_{n=1}^N \left[\frac{1}{\sqrt{\pi}} G_{2,4}^{3,1}\left(g^2\left(\left|\widehat{x}_n^{(1)}\right|\right) \mid_{0,0,\eta-\frac{1}{2},-\eta+\frac{1}{2}}^{0.5,1}\right) \right. \\ \left. - \sqrt{\pi} S_{\eta-\frac{1}{2}}\left(g\left(\left|\widehat{x}_n^{(1)}\right|\right)\right) \cdot G_{2,4}^{4,0}\left(g^2\left(\left|\widehat{x}_n^{(1)}\right|\right) \mid_{0,0,\eta-\frac{1}{2},-\eta+\frac{1}{2}}^{0.5,1}\right) \right] = 0, \quad (\text{B.12}) \end{aligned}$$

where $R_\mu(\cdot) = \frac{K_{\mu+1}(\cdot)}{K_\mu(\cdot)}$, $S_\mu(\cdot) = \frac{I_\mu(\cdot)}{K_\mu(\cdot)}$ with $I_\mu(\cdot)$ the modified Bessel function of the first kind [81, 8.431.1, page 916]

$$I_\mu(v) = \frac{\left(\frac{v}{2}\right)^\mu}{\sqrt{\pi}\Gamma\left(\mu + \frac{1}{2}\right)} \int_{-1}^1 (1-t^2)^{\mu-\frac{1}{2}} e^{-vt} dt, \quad (\text{B.13})$$

and $G_{p,q}^{m,n}(\cdot | \cdot)$ is the Meijer G -function [81, 9.301, page 1032]

$$G_{p,q}^{m,n}\left(z \mid \begin{matrix} a_1, \dots, a_p \\ b_1, \dots, b_q \end{matrix}\right) = \frac{1}{2\pi i} \int_L \frac{\prod_{l=1}^m \Gamma(b_l - s) \prod_{l=1}^n \Gamma(1 - a_l + s)}{\prod_{l=m}^{q-1} \Gamma(1 - b_{l+1} + s) \prod_{l=n}^{p-1} \Gamma(a_{l+1} - s)} z^s ds, \quad (\text{B.14})$$

in which the integration path L separates the poles of the factors $\Gamma(b_l - s)$ from those of the factors $\Gamma(1 - a_l + s)$, m and n are integers such that $0 \leq m \leq q$ and $0 \leq n \leq p$, and none of $a_k - b_j$ is a positive integer when $0 \leq k \leq n$ and $1 \leq j \leq m$.

Appendix C

French Summary

"Une image vaut plus que dix mille mots". La force de ce vieil adage s'est conservée au fil des siècles et jusqu'à aujourd'hui. Au début, le terme *peinture* était utilisé pour décrire des événements du monde réel au lieu de *mille mots*. Ce rôle a été assumé par la *photographie* depuis l'invention des appareils photographiques d'images fixes au 19^e siècle. À cette époque, "voir c'est croire" car les images sont capturées par des dispositifs d'acquisition analogiques et stockées sur des films photographiques. La manipulation des images sur de tels supports physiques est extrêmement difficile et requiert les compétences particulières de professionnels. Par conséquent, une image imprimée dans un journal était communément acceptée comme une certification de l'information. Au début du 21^e siècle, la révolution numérique a radicalement changé la façon dont une image est imprimée. Cela a changé non seulement la façon dont une image est acquise, mais aussi la manière dont les contenus des images sont stockés et transmis. Les ordinateurs portables, clés USB, cartes mémoires, disques compacts sont quelques exemples de dispositifs personnels capables de stocker des images numériques, tandis que les réseaux sociaux sont des services puissants permettant de stocker et de partager des contenus numériques partout et à tout moment, grâce à des appareils numériques connectés à Internet. Grâce à ces outils du quotidien, presque tout le monde dans notre vie moderne est capable d'enregistrer, de stocker et de partager un grand nombre d'images numériques. En outre, la diffusion de logiciels d'édition d'images puissants et peu coûteux a rendu la modification des images numériques plus facile. Cela implique que les images ne sont plus quelque chose d'immuable, mais plutôt comme un *organisme vivant* qui évolue au fil du temps au moyen de diverses opérations. En conséquence, nous en sommes venus socialement à comprendre les événements d'une manière beaucoup plus visuelle : "L'image n'est plus que ce que l'on voit" de sorte qu'apparemment "voir n'est plus croire".

Il importe peu que la manipulation de l'image ne serve qu'à des objectifs sains comme le diagnostic médical, la satire et l'embellissement. Malheureusement, les images manipulées ont été utilisées à des fins malveillantes liées à tous les aspects de la vie humaine tels que la politique, l'écologie, la race et le sexe. En 1997, 58 touristes innocents ont été tués lors d'un attentat terroriste au temple d'Hatchepsout à Louxor en Égypte (voir la figure C.1a). Le journal suisse Blick a rapporté l'événement en modifiant la couleur de l'eau pour qu'elle ressemble à du sang s'écoulant du temple (voir la figure C.1b), afin que le public puisse comprendre la brutalité de l'attaque. Le 09 juillet 2008, les journaux "Palm Beach

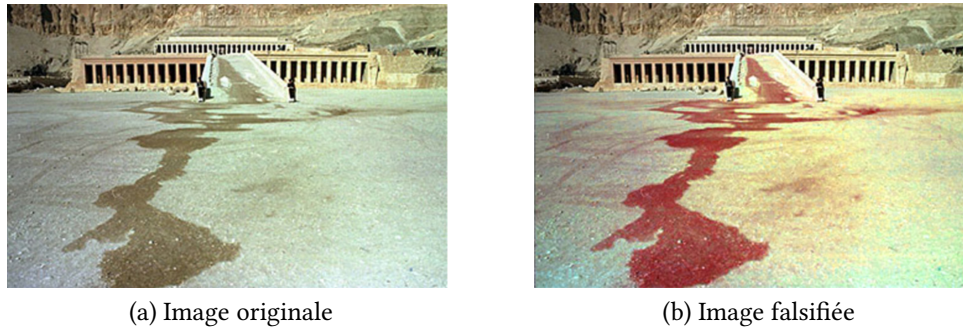


Figure C.1: Événement de l'attaque terroriste au temple d'Hatchepsout à Louxor en Égypte

Post" et le "Chicago Tribune" ont publié en première page une photo décrivant un test de missile iranien (voir la figure C.2a). Après la publication, il a été révélé que le deuxième missile à partir de la droite avait été cloné afin de dissimuler un missile au sol qui n'avait pas été tiré (voir la figure C.2b). La photographie authentique (voir la figure C.2c) est apparue un jour plus tard confirmant une nouvelle fois que la photographie publiée était en fait falsifiée. En juin 2010, la couverture de



Figure C.2: Photographie originale

"The Economist" montrait le président Barack Obama seul sur une plage de Louisiane examinant les conséquences de la marée noire de British Petroleum dans le golfe du Mexique (voir la figure C.3a). Sous le titre "Obama contre BP", le président semble réfléchir à la manière de gérer cette catastrophe politique et écologique. Cependant, en juillet 2010, le blog *New York Times Media Decoder* a révélé que le président n'était pas seul (voir la figure C.3b). Les événements ci-dessus montrent clairement que les images falsifiées sont devenues un problème majeur pour la société. En fait, les images falsifiées apparaissent de plus en plus fréquemment et de manière sophistiquée avec l'avancée croissante de nouvelles techniques d'intelligence artificielle pour lesquelles aucun savoir-faire technique spécifique n'est requis de la part des utilisateurs. Cette situation grave conduit à la nécessité de stratégies et de méthodes qui permettent de vérifier l'authenticité des images numériques.

C.1 Cadre de recherche

La criminalistique des images numériques est apparue comme un domaine de recherche indispensable pour restaurer une certaine confiance dans les images numériques. De manière générale, elle se



(a) Image falsifiée



(b) Image originale

Figure C.3: Événement du président Barack Obama seul sur une plage de Louisiane

consacre à chercher des réponses à deux grandes questions :

1. *d'où vient l'image numérique?*
2. *est-ce que l'image numérique représentante la scène originale capturée?*

La première question est étroitement liée au problème de l'identification de l'origine de l'image avec comme principal objectif la spécification du système d'acquisition. La deuxième question concerne le problème de l'*intégrité du contenu de l'image* où il s'agit de déterminer si une image est falsifiée, quelle manipulation a été effectuée sur l'image ou quelle région de l'image a été modifiée. Dans cette thèse, nous nous concentrons plus particulièrement sur le développement de méthodes de détection et de localisation de faux dans des images numériques.

La littérature relative à la détection et à la localisation de faux a été abordée selon deux approches : (i) *active*, et (ii) *passive*. Par approche active, certaines informations authentiques prédéfinies telles que la signature ou le filigrane intégrés dans les images numériques sont nécessaires pour examiner leur véracité. Cependant, ces informations constituent une limite à cette approche, car elles augmentent le coût de production. Pour remédier à cet inconvénient, l'approche passive propose d'exploiter les traces intrinsèques à l'acquisition de l'image ou certaines traces spécifiques laissées par les contrefaçons pour distinguer les images falsifiées des images naturelles. Cette approche est évidemment adaptée à la plupart des situations pratiques où seule l'image suspecte est disponible. Plus précisément, nous pouvons classer l'approche passive pour la détection et la localisation des falsifications en deux principaux courants. Le premier courant recommande d'extraire des caractéristiques des images suspectes et d'appliquer des techniques d'apprentissage pour apprendre automatiquement des caractéristiques discriminantes [12]. En tenant compte de diverses caractéristiques dans l'apprentissage, cette approche permet de concevoir des outils universels capables de faire face à différents types d'attaques malveillantes [44]. Évidemment, ses performances et sa robustesse dépendent fortement de la qualité et de la quantité des données considérées. Comme alternative, le deuxième courant exploite les incohérences dans la modélisation statistique des images numériques pour détecter les attaques malveillantes (voir [63, 77]). Elle ne nécessite pas une grande quantité de données comme pour la première approche, mais une connaissance plus approfondie du processus d'acquisition des images et des attaques.

C.2 Plan de le thèse

Cette thèse se concentre sur la modélisation statistique pour la détection de la falsification des images numériques. Sa structure générale est organisée comme suit :

- Le chapitre 1 présente le contexte général de la recherche, le problème à traiter, ainsi que la méthodologie employée.
- Le chapitre 2 donne une vue d'ensemble de la détection passive basée sur la modélisation du processus de traitement des images numériques. Nous commençons par une description des principales étapes du traitement de l'image dans un appareil photographique depuis la scène de l'image jusqu'à l'image finale au format JPEG (Joint Photographic Experts Group). Ensuite, nous présentons et illustrons les techniques courantes de falsification d'images numériques. Enfin, nous classons les méthodes de détection de la falsification d'images en fonction des traces laissées par les différentes étapes de traitement.
- Le chapitre 3 se concentre sur l'authentification des images numériques et la localisation de faux en utilisant les artefacts du dématricage. Après avoir analysé l'évolution des traces de dématricage dans le processus d'acquisition de l'image, une statistique robuste caractérisant les images numériques dématricées est d'abord développée sur la base du bruit présent dans le canal vert. Ensuite, un modèle de mélange de lois normales est proposé pour décrire la distribution de probabilité des caractéristiques statistiques pour les images originales et trafiquées. Ainsi, un test de normalité peut être utilisé pour authentifier automatiquement les images falsifiées. Enfin, un algorithme pénalisé est utilisé pour localiser les zones falsifiées. Une étude comparative sur quatre ensembles de données bien connus montre que l'algorithme développé offre de meilleures performances et une meilleure robustesse que les algorithmes existants du même type.
- Le chapitre 4 traite de la modélisation statistique des coefficients de la transformée en cosinus discrète des images JPEG authentiques et falsifiées et de son application à la localisation des falsifications.
- Le chapitre 5 développe un détecteur efficace pour distinguer une image au format TIFF (Tagged Image File Format) rééchantillonnée d'une image TIFF originale. À cette fin, nous proposons un modèle statistique pour les images TIFF rééchantillonnées en analysant le processus de traitement complet d'une image RAW à une image TIFF rééchantillonnée. Ensuite, nous formulons le problème de détection comme un test de rapport de vraisemblance entre les modèles des images originales et rééchantillonnées. La puissance du test est analytiquement évaluée dans le contexte où tous les paramètres du modèle des images TIFF sont inconnus. De nombreuses expériences numériques décrivent les performances du détecteur.
- Le chapitre 6 décrit les principaux résultats et présente quelques perspectives sur les travaux futurs dans le domaine de la criminalistique des images numériques.

C.3 Processus de traitement des images numériques

L'analyse de la falsification des images numériques nécessite une connaissance de la création de ces images. Cette section fournit une meilleure compréhension sur la formation des images des caméras numériques. Bien que plusieurs appareils puissent être utilisés pour l'imagerie numérique tels qu'un scanner, nous limitons notre étude aux appareils photographiques. L'ensemble du processus de traitement d'une image numérique, de la capture de la lumière jusqu'au stockage de l'image est illustré par la figure C.4. Dans ce qui suit, nous divisons ce processus de traitement en trois parties en fonction

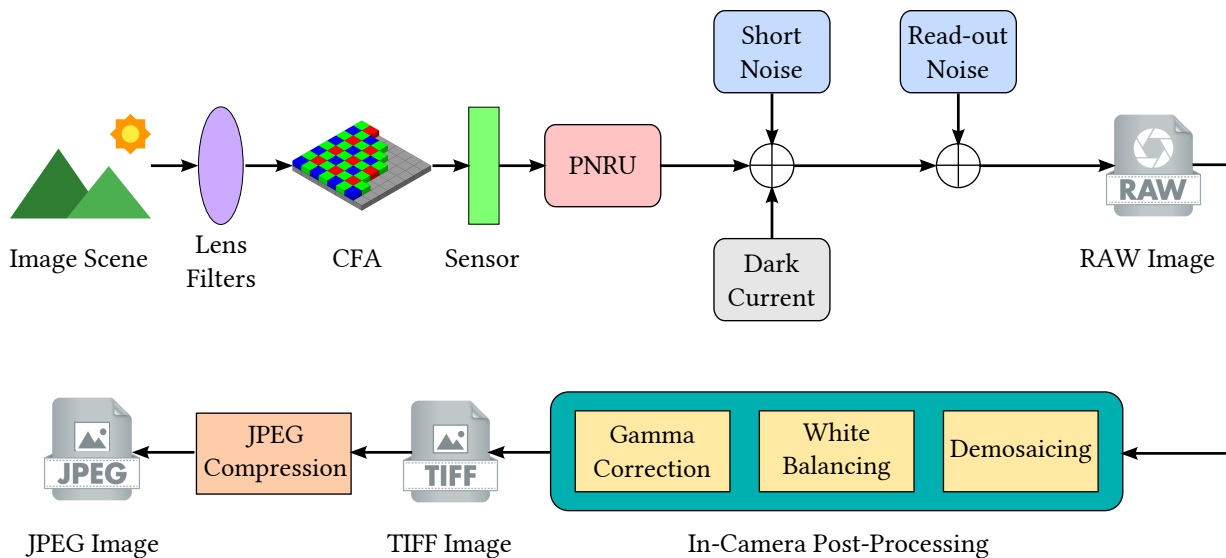


Figure C.4: Illustration du processus d'acquisition d'un appareil photographique numérique

des formats de stockage d'une image: (i) de la scène à l'image brute, (ii) de l'image brute à l'image TIFF, et (iii) de l'image TIFF à l'image JPEG. Pour chaque partie, nous analysons les principales étapes du traitement et nous introduisons les principaux modèles statistiques de l'image.

C.3.1 De la scène à l'image brute

Les appareils numériques imitent le système visuel humain. Ils sont constitués d'un objectif, de filtres d'échantillonnage, d'un réseau de filtres colorés (CFA), d'un capteur d'image et d'un processeur d'image numérique[1]. Le module optique est essentiellement composé d'un objectif et des mécanismes permettant de contrôler l'exposition, la mise au point et la stabilisation de l'image pour collecter et piloter la lumière provenant de la scène. Une fois que la lumière entre dans la caméra via l'objectif, elle passe à travers une combinaison de filtres qui comprend des filtres infrarouge et des filtres anti-crénelage pour garantir une qualité visible maximale. La lumière est ensuite focalisée sur le capteur d'image, un ensemble de rangées et de colonnes d'éléments de détection de la lumière appelés pixels. Chaque élément de détection de la lumière du capteur intègre la lumière incidente sur tout le spectre et obtient un signal électrique représentant la scène. Comme chaque élément de capteur d'image est essentiellement monochromatique, une façon simple de capturer des images en couleur est de séparer chaque composante de couleurs.

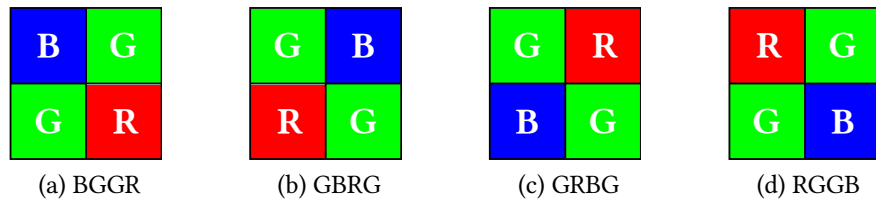


Figure C.5: Four possible configurations of the Bayer pattern

Parmi les nombreux motifs CFA, celui conçu par Bayer [14] est le plus utilisé dans les caméras numériques. Il mesure la composante verte sur une grille *en quinconce* et les composantes rouge et bleue sur des grilles *rectangulaires*. Le taux d'échantillonnage plus élevé pour le canal vert permet de mieux capturer la luminance de la lumière. Le motif de Bayer a lui-même quatre configurations possibles comme le montre la figure C.5. Nous cherchons une représentation mathématique \mathbf{V} d'une image brute capturée avec une configuration \mathbf{V} du motif de Bayer. Comme l'illustre la figure C.6, l'image brute \mathbf{V} est de type monocanal car la configuration de Bayer \mathbf{B} permet à chaque pixel de n'enregistrer qu'un seul canal de couleur c parmi le rouge (r), le vert (g) ou le bleu (b). Par conséquent,

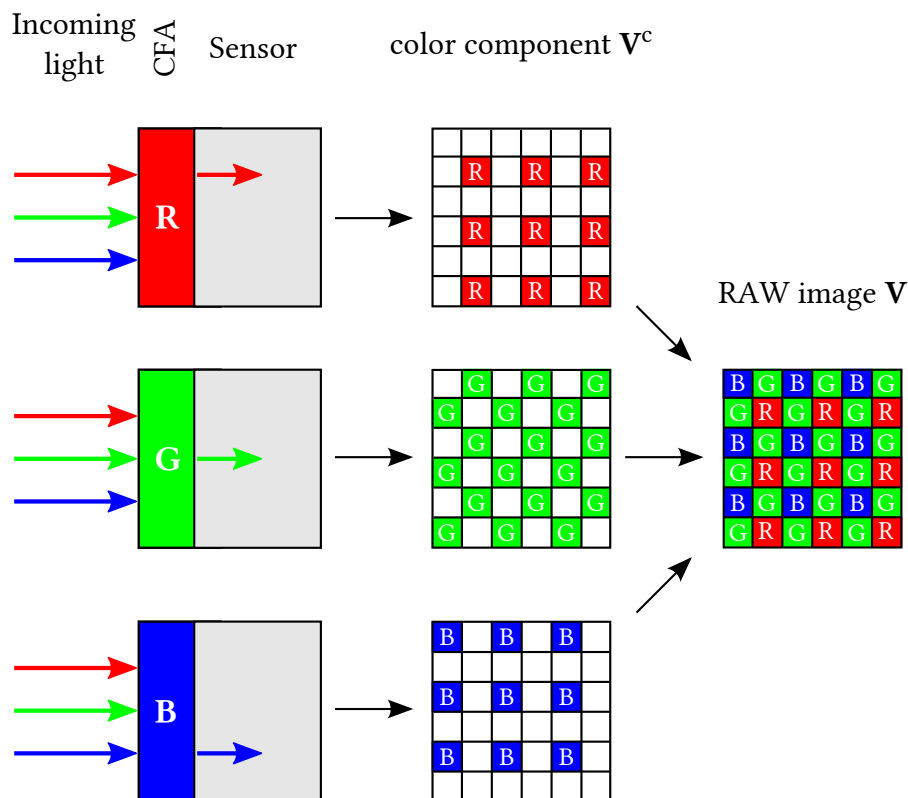


Figure C.6: Processus d'obtention d'une image brute

nous pouvons exprimer \mathbf{V} comme une matrice bidimensionnelle de taille $M \times N$ dont la valeur du pixel

à l'emplacement (m, n) de la composante couleur V^c est donnée par

$$v_{m,n}^c = \begin{cases} v_{m,n} & \text{if } \mathbf{B}_{m,n} = c, \\ 0 & \text{otherwise,} \end{cases} \quad (\text{C.1})$$

où $1 \leq m \leq M$, $1 \leq n \leq N$, et $\mathbf{B}_{m,n}$ désigne l'emplacement (m, n) de la configuration du filtre de Bayer.

Le processus d'acquisition des images brutes est perturbé par plusieurs sources de bruit dont le bruit de grenaille, le bruit de courant d'obscurité, le bruit de lecture et le bruit de non-uniformité de la réponse photographique (PRNU). Le dernier bruit est le plus importants (voir la figure C.4). Le bruit de grenaille, également connu sous le nom de bruit de Poisson, trouve son origine dans la nature quantique de la lumière. Le bruit du courant d'obscurité, également appelé bruit à motif fixe, est généré par l'énergie thermique en l'absence de lumière. Le bruit de lecture englobe tous les bruits électroniques intervenant dans la chaîne d'acquisition. Enfin, le bruit PRNU représente les différences de réponse des pixels à la lumière incidente due aux imperfections du processus de fabrication des capteurs (par exemple l'hétérogénéité des plaquettes de silicium). En tenant compte de ces sources de bruit, nous pouvons exprimer le pixel de la manière suivante $v_{m,n}$ dans (C.1) comme suit

$$v_{m,n} = \mu_{v_{m,n}} + \epsilon_{v_{m,n}}, \quad (\text{C.2})$$

où $\mu_{v_{m,n}}$ représente l'espérance de $v_{m,n}$ en l'absence de bruit et $\epsilon_{v_{m,n}}$ représente toutes les sources de bruit. Alors que l'espérance $\mu_{v_{m,n}}$ est simplement vue comme un signal déterministe et ne doit pas être modélisée. Les modèles de bruit pour les images RAW peuvent être classés en deux classes dans la littérature : (i) *indépendant du signal*, et (ii) *dépendant du signal*. Si la première suppose la stationnarité du bruit dans l'ensemble de l'image, la seconde prend en compte la dépendance proportionnelle de la variance du bruit par rapport au signal. Le modèle bien connu de bruit blanc additif gaussien (voir par exemple [22]) est un exemple typique de la classe de bruit indépendant du signal. La classe de bruits dépendants du signal comprend le modèle Poissonien [119], le modèle de bruit Poisson-Gaussien [141, 149] et le modèle de bruit hétéroscédastique [67, 205]. Par rapport à la première classe, la seconde est plus précise mais plus difficile à traiter.

C.3.2 De l'image brute à l'image TIFF

Pour rendre une image TIFF non compressée en couleur à partir de l'image brute, plusieurs opérations de post-traitement sont effectuées où le *dématriçage*, *l'équilibrage des blancs* et *la correction gamma* sont les plus importantes.

Le *dématriçage*, également connu sous le nom d'*interpolation CFA*, permet de combler les pixels manquants dus à l'échantillonnage CFA en utilisant leur voisinage [185]. En général, on peut classer les algorithmes de *dématriçage* en deux classes : (i) *non adaptatif* et (ii) *adaptatif*. Les algorithmes non adaptatifs utilisent la même technique d'interpolation pour tous les pixels, de sorte que, dans la plupart des cas, ils peuvent être simplement définis par un seul schéma pour chaque canal de couleur c (voir le tableau C.1). Par conséquent, la composante de couleur \mathbf{X}_D^c de l'image *dématriçée* \mathbf{X}_D peut

name	Canaux rouge et bleu	Canal vert
bi-linear	$\mathbf{K}^r = \mathbf{K}^b = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	$\mathbf{K}^g = \frac{1}{4} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$
bi-cubic	$\mathbf{K}^r = \mathbf{K}^b = \frac{1}{256} \begin{bmatrix} 1 & 0 & -9 & -16 & -9 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -9 & 0 & 81 & 144 & 81 & 0 & -9 \\ -16 & 0 & 144 & 256 & 144 & 0 & -16 \\ -6 & 0 & 81 & 144 & 81 & 0 & -6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -9 & -16 & -9 & 0 & 1 \end{bmatrix}$	$\mathbf{K}^g = \frac{1}{256} \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -9 & 0 & -9 & 0 & 0 \\ 0 & -9 & 0 & 81 & 0 & -9 & 0 \\ 1 & 0 & 81 & 256 & 81 & 0 & 1 \\ 0 & -9 & 0 & 81 & 0 & -9 & 0 \\ 0 & 0 & -9 & 0 & -9 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$

Table C.1: Exemples de noyaux d'interpolation utilisés dans les algorithmes de dématricage non adaptatifs

être calculée à partir de la composante brute \mathbf{V}^c comme suit:

$$\mathbf{X}_D^c = \mathbf{K}^c * \mathbf{V}^c, \quad (\text{C.3})$$

où $*$ désigne l'opération de convolution bidimensionnelle. Les algorithmes de dématricage non adaptatifs donnent des résultats satisfaisants dans les régions lisses, mais échouent généralement dans les régions texturées et les bords. Les algorithmes adaptatifs peuvent résoudre ce problème. Ils prennent en compte les informations sur les bords, ainsi que la corrélation inter-canaux pour trouver un ensemble approprié de coefficients qui permet de minimiser l'erreur d'interpolation globale ((voir [42, 210])). Cela rend les algorithmes adaptatifs plus précis que les algorithmes non adaptatifs, au prix d'un calcul plus important.

Après le dématricage, il faut procéder à l'équilibrage des blancs pour supprimer les teintes irréalistes [186]. En effet, en raison de la différence de température des sources lumineuses, un décalage du spectre de réflexion de l'objet par rapport à sa vraie couleur peut se produire. Ce décalage fait que l'objet apparaît de couleur différente lorsqu'il est éclairé par différentes sources lumineuses. Le réglage de la balance des blancs est donc mis en oeuvre dans les appareils numériques pour compenser ce déséquilibre d'éclairage. Fondamentalement, il est effectué en multipliant les pixels de chaque canal de couleur \mathbf{X}_D^c par un facteur de gain différent g_W^c pour obtenir une image équilibrée au niveau du blanc \mathbf{X}_W .

$$\mathbf{X}_W^c = g_W^c \cdot \mathbf{X}_D^c, \quad (\text{C.4})$$

où \mathbf{X}_W^c désigne la composante de \mathbf{X}_W associée au canal de couleur c . La connaissance préalable des sources lumineuses est essentielle pour sélectionner les gains appropriés g_W^c . En l'absence de cette connaissance, g_W^c peut être estimé par plusieurs algorithmes décrits dans [121]. A titre d'exemple, nous présentons ci-après un algorithme simple basé sur l'hypothèse *mode gris: la réflectance moyenne d'une scène est achromatique*. Cela implique que les valeurs moyennes des trois canaux de couleur

dans une scène donnée sont à peu près égales de sorte que

$$\bar{x}_W^r = \bar{x}_W^g = \bar{x}_W^b, \quad (\text{C.5})$$

où \bar{x}_W^c , En utilisant (C.4) et (C.5), il en résulte que

$$g_W^r \cdot \bar{x}_D^r = g_W^g \cdot \bar{x}_D^g = g_W^b \cdot \bar{x}_D^b, \quad (\text{C.6})$$

où \bar{x}_D^c représente l'intensité moyenne de \mathbf{X}_D^c , calculée de la même manière que (2.6). Puisque l'oeil humain est plus sensible à la couleur verte, l'algorithme propose de conserver le le canal vert tel qu'il est, donc

$$g_W^b = 1. \quad (\text{C.7})$$

Par conséquent, les gains des autres canaux de couleur sont dérivés de (C.6) comme suit

$$g_W^r = \frac{\bar{x}_D^g}{\bar{x}_D^r} \quad \text{and} \quad g_W^g = \frac{\bar{x}_D^g}{\bar{x}_D^g}. \quad (\text{C.8})$$

En introduisant (C.7) et (C.8) dans (C.4), nous obtenons finalement l'image équilibrée en blanc \mathbf{X}_W .

A ce stade, l'intensité du pixel ne présente que la caractéristique linéaire de l'intensité de l'image brute [53]. Cette linéarité ne correspond pas à la plupart des dispositifs d'affichage dont la fonction de transfert représente une relation entre la luminance L et la tension U

$$L = V^\gamma, \quad (\text{C.9})$$

où γ est une constante spécifique à chaque dispositif. En particulier, $\gamma = 2,2$ pour le moniteur à tube cathodique. La correction gamma est nécessaire pour rendre la luminance dans un domaine perceptiblement uniforme. Il s'agit simplement de l'inverse de (C.9) en appliquant à chaque valeur de pixels de \mathbf{X}_W une valeur telle que

$$x_{m,n}^c = \left(x_{W,m,n}^c \right)^{\frac{1}{\gamma}}, \quad (\text{C.10})$$

où $x_{m,n}^c$ désigne le pixel situé à l'emplacement (m, n) , de la composante \mathbf{X}^c de l'image TIFF.

A la sortie du processus précédent, nous obtenons une image TIFF non compressée, en couleur, \mathbf{X} , caractérisée par une matrice tridimensionnelle de taille $M \times N \times 3$ (voir figure C.7).

C.3.3 De l'image TIFF à l'image JPEG

En raison de sa haute qualité, l'image TIFF n'est pas vraiment adaptée pour le stockage ou la transmission, surtout en grand nombre. C'est pourquoi la plupart des appareils photographiques numériques utilisent un algorithme de compression pour réduire la taille des données de l'image. Un tel algorithme tente d'éliminer les informations de l'image qui ne sont pas visuellement significatives. Parmi les nombreux algorithmes de compression avec perte (voir par exemple [92], la norme JPEG (Joint Photographic Experts Group) [213] est la plus populaire grâce à son bon compromis entre la qualité

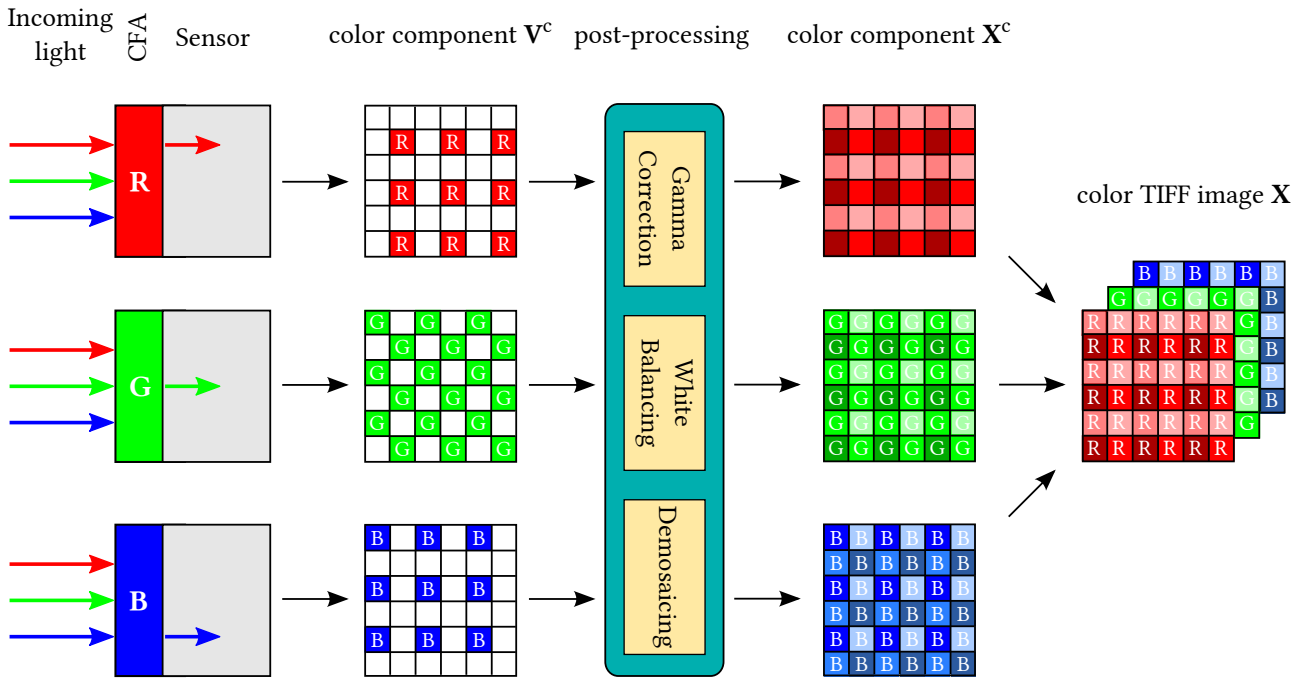


Figure C.7: Processus de formation d'une image au format TIFF

visuelle et la taille de l'image. Comme le montre la figure C.8, une chaîne de compression JPEG typique consiste en quatre étapes fondamentales: (i) la transformation de la couleur, (ii) la transformée en cosinus discret (iii), la quantification, et (iv) le codage entropique.

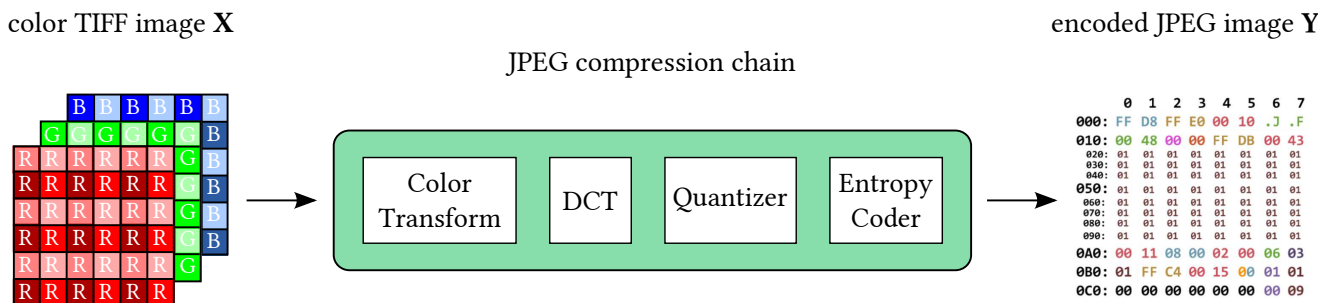


Figure C.8: Les étapes fondamentales dans une chaîne de compression JPEG

C.4 Localisation des falsifications à partir du dématricage

Dans le processus d'acquisition d'une caméra numérique, le dématricage sert à reconstruire une image en couleurs à partir des données échantillonnées superposées à un réseau de filtres colorés (voir la figure C.4). Bien que la plupart des appareils photographiques partagent un processus d'acquisition commun, chaque étape est personnalisée en fonction des choix du fabricant. Les traces laissées par le dématricage sont spécifiques aux différentes marques et/ou modèles de caméras. Ces traces sont totalement absentes d'une image lorsque l'image est susceptible d'être falsifiée. En outre, étant donné que le dématricage est une opération située en amont dans le processus d'acquisition, ses traces sont

résistantes aux manipulations post-caméra telles que le clonage, l'épissage, le redimensionnement, etc... Par conséquent, les traces du dématricage pourraient être une preuve robuste et puissante pour évaluer la crédibilité des images numériques.

Analyse bibliographique

Nous pouvons classer les travaux de médecine légale qui considèrent les traces de dématricage en deux classes principales. La première consiste en des algorithmes visant à estimer, à partir d'une image numérique, le motif CFA et/ou l'algorithme de dématricage inhérent aux appareils photographiques. La seconde caractérise les algorithmes utilisés pour évaluer la présence/absence d'artefacts générés par les opérations de dématricage. Dans la suite de ce document, nous détaillons ces 2 classes.

Première classe

Les travaux [38, 96, 108, 193, 202] présentent des méthodes clés pour l'identification du modèle CFA. Dans [108], le motif CFA de Bayer est identifié en réduisant au minimum la différence entre le signal brut du capteur et le signal inverse dématricé. Dans [38], l'identification est effectuée via un algorithme développé à partir de l'observation que la valeur des échantillons de couleur interpolés se situe toujours entre les valeurs minimale et maximale de leurs voisins. Une autre méthode consiste à calculer le rapport entre la variance moyenne du bruit des pixels interpolés et des pixels acquis pour tous les motifs candidats possibles du CFA [202]. Plus récemment, des blocs de couleur ont été proposés comme moyen permettant d'estimer la configuration du CFA dans [96, 193].

Au-delà de l'identification du modèle CFA, des travaux se concentrent plus particulièrement sur l'estimation des algorithmes de dématricage. Dans [174], un algorithme de maximisation de l'espérance (EM) est utilisé pour estimer les coefficients des noyaux d'interpolation linéaire. Une amélioration est apportée dans [15] en combinant l'algorithme EM avec un spectre dérivé du second ordre [73] pour obtenir des coefficients d'interpolation en régions lisses et non lisses séparément. Dans [27], une méthode précise basée sur un modèle de dérivées partielles du second ordre est proposé pour récupérer les formules de dématricage. La méthode permet de prendre en compte la corrélation entre les trois canaux de couleur de l'image.

Étant donné que le motif CFA et/ou l'algorithme de démosaïquage sont spécifiques à chaque modèle de caméras, ils peuvent être utilisés comme preuves pour la détection et la localisation des falsifications d'images. Par exemple, dans [28, 200], les incohérences entre les coefficients de dématricage sont exploitées pour vérifier si le modèle CFA et l'algorithme de dématricage sont compatibles. En remarquant que le modèle CFA est changé si la couleur de l'image est modifiée, les auteurs de [39] ont conçu un algorithme avancé de comptage de valeurs intermédiaires pour mesurer le changement du motif CFA, et donc de localiser l'étendue de la modification de la couleur dans les images numériques. Nous notons également que le motif CFA estimé et/ou l'algorithme de dématricage sont utilisés non seulement pour évaluer la crédibilité du contenu des images numériques, mais aussi pour identifier la source de la caméra (voir [75]). Cependant, ce dernier point n'entre pas dans le cadre de notre étude.

Deuxième classe

La deuxième classe repose sur des artefacts périodiques causés par le dématricage. En effet, les capteurs d'images d'un CFA sont généralement organisés périodiquement [153]. En outre, de nombreux algorithmes de dématricage se comportent comme un processus de filtrage où les signaux manquants sont interpolés en appliquant périodiquement un noyau d'interpolation aux signaux acquis (voir [174]). Par conséquent, la périodicité des artefacts de dématricage est intrinsèque aux photographies numériques et peut contribuer à l'authentification des images. Inspirés par cette idée, Popescu et Farid construisent [174] une carte de probabilité pour exprimer le modèle périodique de pixels corrélés. Lorsque des pixels interpolés sont présents, la périodicité est clairement visible dans le domaine de Fourier. Une telle analyse peut être appliquée à différentes zones de l'image pour détecter la présence de d'une altération locale, mais la taille de la zone ne doit pas être inférieure à 256×256 pour garantir la précision des résultats. En observant que la variance de la dérivée seconde des images interpolées est périodique [73], Gallagher et Chen utilisent dans [74] un filtrage passe-haut et une analyse de Fourier pour calculer la fréquence périodique. Les régions falsifiées d'une image auront une périodicité différente que le reste. Cette méthode est applicable pour des blocs d'images de taille 64×64 ou plus. Une autre méthode basée sur l'analyse de la périodicité de la variance des erreurs de prédiction a été récemment proposée par Li *et al.* dans [134]. En supposant que les erreurs d'interpolation sont distribuées de manière gaussienne, une carte de probabilité est dérivée selon la règle de Bayes. Une analyse avec la transformée de Fourier discrète bidimensionnelle permet de capturer la périodicité. Li *et al.* montrent que cette méthode peut être appliquée pour la détection de la falsification locale avec des blocs de taille allant jusqu'à 32×32 .

Les méthodes de détection de la falsification d'images numériques sont également basées sur les différences entre la distribution des pixels acquis et interpolés. Dans [55], Dirik et Memon reconnaissent que la nature passe-bas du dématricage rend la variance du bruit du capteur plus importante. En conséquence, les artefacts de dématricage peuvent être mesurés par un rapport des variances du bruit entre les pixels interpolés et acquis. Si ce rapport est proche de 1, une altération a été réalisée sur l'image. La méthode présente une bonne performance pour les blocs d'images de taille supérieure à 96×96 . Partageant la même idée, Ferrara *et al.* [63] effectuent une analyse fine des artefacts du dématricage et proposent une fonction permettant de mesurer la présence du dématricage. Alors que les travaux ci-dessus considèrent les caractéristiques spatiales des artefacts de démosaïquage, González-Fernández *et al.* [79] s'intéressent plutôt à leur spectre. En effet, en calculant la probabilité de chaque pixel d'être interpolé puis en appliquant la transformée en cosinus discrète (DCT) sur de petits blocs de la carte de probabilité, la présence/absence d'artefacts de dématricage. Les expériences montrent que la méthode est fiable pour des blocs de taille 16×16 .

Remarques communes

Comme le dématricage est une opération en amont du processus d'acquisition des caméras numériques, ses traces sont indépendantes des falsifications. Par conséquent, les méthodes basées sur les traces de dématricage ne ciblent pas une opération de falsification spécifique, mais plutôt une variété d'opérations.

Malgré cet avantage, les traces de dématricage sont facilement détruites par la compression JPEG, même avec des niveaux de qualité très élevés. C'est pourquoi ces méthodes sont adaptées aux photographies non compressées ou peu compressées. Il s'agit d'une limitation courante et presque inévitable des méthodes basées sur les traces de dématricage. De plus, très souvent, un seuil de comparaison est nécessaire pour détecter ou localiser les faux [55, 63, 134]. En pratique, le choix d'un tel seuil n'est pas facile et peut avoir une grande influence sur la robustesse des méthodes. En outre, les traces de dématricage utilisées (c'est-à-dire le modèle CFA, l'algorithme de dématricage, la périodicité des artefacts de dématricage, les différences dans la distribution des pixels acquis et interpolés, etc..) sont étroitement corrélées. Cependant, la classification ci-dessus des travaux connexes implique qu'ils sont traités séparément dans la plupart des algorithmes existants. L'utilisation conjointe de ces traces pourrait améliorer les performances de la détection et de la localisation de la falsification d'images.

Problématiques et contributions

Les deux classes d'algorithmes utilisant des traces de dématricages ont été étudiées séparément dans la littérature. Ici, nous cherchons à les combiner dans un algorithme unifié afin d'améliorer les performances d'authentification de l'image et la localisation de la contrefaçon. Le résultat est un algorithme hybride composé de l'identification du motif CFA, de l'estimation de l'algorithme de dématricages et de l'analyse des artefacts. En visant une détection fine, nous avons adopté le schéma d'identification proposé dans [199] pour révéler le motif CFA et le noyau d'interpolation et une analyse locale similaire à celle de [63] pour déterminer les artefacts de dématricages des résidus de prédiction.

1. Nous analysons théoriquement comment la moyenne et la variance des résidus de prédiction dans les signaux interpolés et acquis évoluent dans le processus d'acquisition d'un appareil photographique, du format RAW au format JPEG. Ceci constitue la base pour construire des caractéristiques d'intérêt. En outre, l'analyse permet d'expliquer en partie pourquoi l'approche basée sur les traces de dématricage est moins efficace avec les images compressées au format JPEG.
2. Dans la plupart des algorithmes de dématricage basés sur les traces (voir [63, 134, 174, 194]), les résidus de prédiction sont donnés à partir des valeurs des pixels. Ici, en partitionnant un pixel en 2 parties comportant le contenu et le bruit, nous réalisons que le dématricage se comporte de la même manière pour le contenu et le bruit. Cependant, en raison de la faible énergie du bruit, les traces du dématricage sont plus visibles dans le bruit. C'est pourquoi les caractéristiques utilisées dans l'algorithme sont extraites des résidus de bruit plutôt que des résidus du contenu/
3. Comme mentionné dans [63], la présence de bords nets dans les images réduit la qualité des caractéristiques, car elle peut perturber par la corrélation entre les résidus interpolés et acquis. Pour surmonter cet obstacle, nous construisons nos caractéristiques statistiques à partir d'une région lisse du canal vert au lieu de la totalité de l'image. Ainsi, le problème des bords n'est plus un problème majeur.

4. En ce qui concerne l'authentification, après avoir vérifié la distribution normale des caractéristiques dans les images naturelles, nous adoptons la loi normale pour l'authentification.
5. En modélisant la distribution des caractéristiques statistiques dans les images falsifiées comme un mélange de lois normales, nous avons appliqué un algorithme *pénalisé d'espérance-maximisation* (EM) pour localiser les régions falsifiées. Puisque l'algorithme ne nécessite pas de seuil de comparaison, il fournit des résultats de localisation plus robustes que les méthodes traditionnelles [55, 63].

C.5 Modélisation des coefficients DCT d'images JPEG et localisation des contrefaçons

Le format JPEG (Joint Photographic Experts Group) est une norme de compression avec perte de données permettant de trouver un compromis entre la taille de la mémoire et la qualité visuelle pour les images numériques [213]. Elle est aujourd'hui adoptée par défaut dans la plupart des appareils photographiques numériques et les services Web (réseaux sociaux, galeries photos, etc.). Cependant, cette popularité fait également des images JPEG la cible d'attaques malveillantes. Cela nous motive à développer dans ce chapitre un outil d'analyse fiable pour restaurer la confiance dans les images JPEG. Plus particulièrement, nous nous intéressons à la statistique des coefficients de la transformée en cosinus discrète (DCT).

Analyse bibliographique

De nombreux efforts ont été déployés pour construire des modèles statistiques pour les coefficients AC non quantifiés des images naturelles JPEG. Le pionnier est peut-être Pratt, qui a adapté les propriétés statistiques des coefficients AC non quantifiés primaires avec une distribution normale [177, chapitre 10]. Par la suite, divers modèles empiriques ont été proposés et vérifiés par des tests d'adéquation standard. Par exemple, Reininger et Gibson [188] se sont appuyés sur des tests de Kolmogorov-Smirnov (KS) pour confirmer que la distribution de Laplace (LD) est plus appropriée que la distribution normale pour la plupart des images. En se basant sur un test du χ^2 , Müller [157] a affirmé que la distribution normale généralisée (DNG) approxime les statistiques des coefficients AC non quantifiés mieux que la loi normale. En utilisant les mêmes tests d'adéquation, Chang *et al.* a récemment montré dans [30] que la distribution Gamma généralisée est plus performante que la loi normale et la DNG. Le principal inconvénient de ces modèles empiriques ci-dessus est l'absence de fondement mathématique, ce qui fait que leur précision et leur robustesse sont remises en question lorsqu'ils sont appliqués à un large éventail d'images. Pour remédier à cet obstacle, Lam et Goodman ont analysé analytiquement dans [122] les propriétés statistiques des coefficients AC non quantifiés par rapport à la variation de leur variance. Ils ont prouvé que ces coefficients sont modélisés de manière raisonnable par une loi normale lorsque la variance du bloc est fixe. Cependant, comme cette variance est aléatoire pour une image naturelle JPEG, une distribution normale composée devrait plutôt être utilisée. Cette

distribution a été étendue dans [158] en tenant compte des distributions flexibles pour la variance des blocs. Pourtant, les justifications mathématiques pour le modèle de variance d'un bloc manquaient toujours jusqu'à l'apparition de l'article du Thai *et al.* [206]. Ils ont montré que la variance par bloc d'une image naturelle JPEG peut être approximée de manière asymptotique par une distribution Gamma à deux paramètres. Ce modèle est ensuite appliqué respectivement dans [207] et [189] pour identifier le modèle des caméras et estimer le facteur de qualité des images JPEG.

Les modèles statistiques des coefficients AC quantifiés peuvent être développés à partir des coefficients primaires non quantifiés. Pour les images JPEG à compression unique, Qiao *et al.* a caractérisé les coefficients AC quantifiés par une distribution de Laplace dans [178], et l'ont appliqué pour détecter des informations cachées incorporées par l'algorithme JSteg. Un modèle similaire a également été adopté par Stamm et Liu dans [198] pour supprimer les traces laissées par la compression JPEG. D'autres modèles basés peuvent être trouvés dans [45, 206].

Problématiques et contributions

Grâce à l'analyse de la littérature, nous constatons que les modèles statistiques développés pour les coefficients DCT des images JPEG falsifiées se limitent pour la plupart à l'effet de la double quantification. En effet, dans de nombreux scénarios pratiques, les images peuvent être compressées plusieurs fois avant d'être manipulées et enregistrées à nouveau au format JPEG. Par conséquent, notre objectif est d'étendre ces modèles afin de mieux traiter les images JPEG falsifiées soumises à l'effet de quantifications *multiples*. Notre choix est d'étudier les coefficients DCT dans leur forme brute directement plutôt que dans des formes modifiées comme dans certains travaux connexes (voir [109], [166]). Par conséquent, nous suivons le processus de compression JPEG et les effets des opérations de manipulation sur les coefficients DCT afin de développer un cadre de modélisation statistique paramétrique pour la falsification des images. Nous l'appliquons ensuite à la localisation des falsifications. Bien que l'approche de modélisation et la technique de localisation de falsification soient relativement similaires à certains travaux de la littérature (voir, par exemple, [18, 217]), plusieurs améliorations et contributions significatives ont été apportées.

1. Nous proposons un cadre de modélisation généralisé plutôt qu'un modèle concret pour les coefficients DCT des images JPEG falsifiées. Ce cadre permet de dériver récursivement n'importe quel modèle statistique à partir d'une distribution connue des coefficients DCT non quantifiés. De cette façon, un modèle des coefficients DCT primaires non quantifiés est hérité pour construire un modèle statistique pertinent qui s'adapte le mieux à l'histogramme des coefficients DCT quantifiés.
2. Pour décrire la distribution des coefficients DCT des images JPEG falsifiées, la plupart des travaux existants s'appuient sur les modèles de mélange. Néanmoins, aucune explication claire n'a encore été donnée. Par exemple, le nombre de composantes à prendre en compte dans le modèle est encore une question ouverte. Dans ce chapitre, nous avons prouvé de manière analytique que les histogrammes des DCT quantifiés d'images JPEG falsifiées peuvent être correctement approchés par un modèle de mélange à *deux composantes*, indépendamment du nombre de

zones falsifiées dans les images. La précision d'un tel modèle simplifié est évaluée numériquement par la divergence de Kullback-Leibler sur diverses bases de données d'images.

3. En ce qui concerne la technique de localisation de la falsification, nous nous sommes également appuyés sur l'algorithme Expectation-Maximization (EM), comme dans la plupart des travaux connexes. Cependant, au lieu d'estimer tous les paramètres du modèle en même temps, nous avons proposé d'estimer les paramètres de la distribution des composants, l'ensemble des étapes de quantification et les coefficients de mélange séparément. Le premier est estimé à partir des coefficients DCT non quantifiés d'images JPEG altérées [68] par la méthode d'estimation du maximum de vraisemblance (ML), tandis que la seconde est obtenue en appliquant l'algorithme EM aux coefficients DCT quantifiés associés. Cela permet d'accélérer le processus de localisation de la falsification.

C.6 Détecteurs statistiques pour les images TIFF rééchantillonnées

Dans les chapitres 3 et 4, nous avons traité les situations de falsification dans lesquelles *une ou plusieurs parties* d'une image ont été manipulées. Dans ce chapitre, nous considérons que l'image entière est falsifiée avec une intention malveillante. Pour ce type de falsification, le rééchantillonnage (y compris le redimensionnement, la rotation et d'autres opérations linéaires) est une technique courante [61]. Elle est généralement utilisée conjointement avec d'autres techniques de falsification telles que le clonage ou l'épissage afin de créer un aspect visuel parfait. Cependant, quel que soit le degré de sophistication du rééchantillonnage, il laisse toujours deux principales traces similaires à celles laissées par le dématricage (voir le chapitre 3) : (i) la périodicité du signal rééchantillonné et (ii) l'incohérence de la variance entre le signal original et les signaux interpolés). Comme indiqué dans la sous-section 2.4.2.2 du chapitre 2, la plupart des auteurs se sont concentrés sur les artefacts périodiques pour développer leurs détecteurs de rééchantillonnage. Nous désirons profiter des deux traces ci-dessus pour concevoir des détecteurs statistiques basés sur la théorie des tests d'hypothèse et ainsi concevoir dans ce chapitre un test de rapport de vraisemblance généralisé (LRT) généralisé optimal pour les images TIFF rééchantillonnées. Par rapport à l'état de l'art actuel (par exemple [139, 165, 181, 182]), les principales innovations et contributions de notre travail sont de trois ordres:

1. Au lieu d'utiliser une distribution de probabilité arbitraire, nous analysons le processus de traitement complet d'une image brute à une image TIFF rééchantillonnée afin de trouver un modèle statistique approprié pour les images TIFF rééchantillonnées.
2. Sur la base des modèles développés pour les images TIFF originales et rééchantillonnées, nous exploitons les artefacts périodiques inhérents aux pixels rééchantillonnés pour simplifier les modèles. L'incohérence de la variance du bruit est ensuite utilisée comme mesure principale pour distinguer une image TIFF rééchantillonnée d'une image originale. Il s'agit de la différence fondamentale entre notre travail et la plupart des travaux connexes dont les détecteurs

de rééchantillonnage sont basés sur les artefacts périodiques. Il en résulte deux nouveaux détecteurs basés sur le LRT pour la détection du rééchantillonnage.

3. Enfin, nous sommes en mesure de fournir des expressions pour la fonction de puissance des détecteurs proposés et d'en analyser les propriétés.

C.7 Conclusion

Nous avons développé, dans ces travaux de thèse, trois techniques polyvalentes pour détecter les images numériques contrefaites et localiser les différents types d'altération : (i) *les traces de dématricage* [126], (ii) *les traces de compression JPEG* [127, 128], et (iii) *les traces de rééchantillonnage* [125]. Bien que ces techniques soient différentes, elles fonctionnent selon l'hypothèse commune que les manipulations peuvent altérer certaines propriétés statistiques sous-jacentes des images naturelles. Le même processus en deux étapes a été adopté pour chaque technique de détection : (i) analyser et modéliser les caractéristiques statistiques des images authentiques et falsifiées puis (ii) concevoir un détecteur statistique pour différencier les images authentiques et les images falsifiées en estimant les changements statistiques de leurs modèles.

References

- [1] Jim Adams, Ken Parulski, and Kevin Spaulding. Color processing in digital cameras. *IEEE Micro*, 18(6):20–30, 1998.
- [2] Shruti Agarwal and Hany Farid. Photo forensics from jpeg dimples. In *2017 IEEE Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2017.
- [3] Shruti Agarwal and Hany Farid. Photo forensics from rounding artifacts. In *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security*, pages 103–114, 2020.
- [4] Mustafa Al-Ani and Fouad Khelifi. On the spn estimation in image forensics: a systematic empirical evaluation. *IEEE Transactions on Information Forensics and Security*, 12(5):1067–1081, 2016.
- [5] Osamah M Al-Qershi and Bee Ee Khoo. Evaluation of copy-move forgery detection: datasets and evaluation metrics. *Multimedia Tools and Applications*, 77(24):31807–31833, 2018.
- [6] Irene Amerini, Lamberto Ballan, Roberto Caldelli, Alberto Del Bimbo, and Giuseppe Serra. A sift-based forensic method for copy-move attack detection and transformation recovery. *IEEE transactions on information forensics and security*, 6(3):1099–1110, 2011.
- [7] Irene Amerini, Lamberto Ballan, Roberto Caldelli, Alberto Del Bimbo, Luca Del Tongo, and Giuseppe Serra. Copy-move forgery detection and localization by means of robust clustering with j-linkage. *Signal Processing: Image Communication*, 28(6):659–669, 2013.
- [8] Tae Wuk Bae. Image-quality metric system for color filter array evaluation. *PloS one*, 15(5): e0232583, 2020.
- [9] Khosro Bahrami, Alex C Kot, and Jiayuan Fan. A novel approach for partial blur detection and segmentation. In *2013 IEEE International Conference on Multimedia and Expo*, pages 1–6. IEEE, 2013.
- [10] Khosro Bahrami, Alex C Kot, Leida Li, and Haoliang Li. Blurred image splicing localization by exposing blur type inconsistency. *IEEE Transactions on Information Forensics and Security*, 10(5):999–1009, 2015.

- [11] Daniele Baracchi, Dasara Shullani, Massimo Iuliani, Damiano Giani, and Alessandro Piva. Camera obscura: Exploiting in-camera processing for image counter forensics. *Forensic Science International: Digital Investigation*, 38:301213, 2021.
- [12] Mauro Barni, Luca Bondi, Nicolò Bonettini, Paolo Bestagini, Andrea Costanzo, Marco Maggini, Benedetta Tondi, and Stefano Tubaro. Aligned and non-aligned double jpeg detection using convolutional neural networks. *Journal of Visual Communication and Image Representation*, 49:153–163, 2017.
- [13] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.
- [14] Bryce E Bayer. Color imaging array. US Patent 3,971,065, July 20 1976.
- [15] Sevinc Bayram, Husrev T Sencar, and Nasir Memon. Classification of digital camera-models based on demosaicing artifacts. *digital investigation*, 5(1-2):49–59, 2008.
- [16] Sevinc Bayram, Husrev Taha Sencar, and Nasir Memon. An efficient and robust method for detecting copy-move forgery. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1053–1056, 2009.
- [17] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424, 2000.
- [18] Tiziano Bianchi and Alessandro Piva. Image forgery localization via block-grained analysis of jpeg artifacts. *IEEE Transactions on Information Forensics and Security*, 7(3):1003–1017, 2012.
- [19] Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva. Improved dct coefficient analysis for forgery localization in jpeg images. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2444–2447, 2011.
- [20] Gajanan K Birajdar and Vijay H Mankar. Digital image forgery detection using passive techniques: A survey. *Digital investigation*, 10(3):226–245, 2013.
- [21] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.
- [22] Angelo Bosco, Arcangelo Bruna, Giuseppe Messina, and Giuseppe Spampinato. Fast method for noise level estimation and integrated noise reduction. *IEEE Transactions on Consumer Electronics*, 51(3):1028–1033, 2005.
- [23] Zdravko I Botev, Joseph F Grotowski, Dirk P Kroese, et al. Kernel density estimation via diffusion. *The annals of Statistics*, 38(5):2916–2957, 2010.

- [24] Alexia Briassouli, Panagiotis Tsakalides, and Athanasios Stouraitis. Hidden messages in heavy-tails: Dct-domain watermark detection using alpha-stable models. *IEEE Transactions on Multimedia*, 7(4):700–715, 2005.
- [25] Jason Bunk, Jawadul H Bappy, Tajuddin Manhar Mohammed, Lakshmanan Nataraj, Arjuna Flenner, BS Manjunath, Shivkumar Chandrasekaran, Amit K Roy-Chowdhury, and Lawrence Peterson. Detection and localization of image forgeries using resampling features and deep learning. In *Proceedings of IEEE Conference on the Computer Vision and Pattern Recognition Workshops*, pages 1881–1889, 2017.
- [26] Peter J Burt. Fast algorithms for estimating local image properties. *Computer Vision, Graphics, and Image Processing*, 21(3):368–382, 1983.
- [27] Hong Cao and Alex C Kot. Accurate detection of demosaicing regularity for digital image forensics. *IEEE Transactions on Information Forensics and Security*, 4(4):899–910, 2009.
- [28] Hong Cao and Alex C Kot. Manipulation detection on image patches using fusionboost. *IEEE Transactions on Information Forensics and Security*, 7(3):992–1002, 2012.
- [29] Joon-Hyuk Chang, Jong Won Shin, Nam Soo Kim, and Sanjit K Mitra. Image probability distribution based on generalized gamma function. *IEEE Signal Processing Letters*, 12(4):325–328, 2005.
- [30] Joon-Hyuk Chang, Jong Won Shin, Nam Soo Kim, and Sanjit K Mitra. Image probability distribution based on generalized gamma function. *IEEE Signal Processing Letters*, 12(4):325–328, 2005.
- [31] Jiahua Chen, Xianming Tan, and Runchu Zhang. Inference for normal mixtures in mean and variance. *Statistica Sinica*, pages 443–465, 2008.
- [32] Jiahua Chen, Pengfei Li, and Yuejiao Fu. Inference on the order of a normal mixture. *Journal of the American Statistical Association*, 107(499):1096–1105, 2012.
- [33] Jiahua Chen et al. Consistency of the mle under mixture models. *Statistical Science*, 32(1):47–63, 2017.
- [34] Mo Chen, Jessica Fridrich, Miroslav Goljan, and Jan Lukás. Determining image origin and integrity using sensor noise. *IEEE Transactions on Information Forensics and Security*, 3(1):74–90, 2008.
- [35] Zhipeng Chen, Yao Zhao, and Rongrong Ni. Detection of operation chain: Jpeg-resampling-jpeg. *Signal Processing: Image Communication*, 57:8–20, 2017.
- [36] HR Chennamma and Lalitha Rangarajan. Image splicing detection using inherent lens radial distortion. *International Journal of Computer Science Issues*, 7(6):149, 2010.

- [37] Giovanni Chierchia, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. A bayesian-mrf approach for prnu-based image forgery detection. *IEEE Transactions on Information Forensics and Security*, 9(4):554–567, 2014.
- [38] Chang-Hee Choi, Jung-Ho Choi, and Heung-Kyu Lee. Cfa pattern identification of digital cameras using intermediate value counting. In *Proceedings of the thirteenth ACM multimedia workshop on Multimedia and security*, pages 21–26. ACM, 2011.
- [39] Chang-Hee Choi, Hae-Yeoun Lee, and Heung-Kyu Lee. Estimation of color modification in digital images by cfa pattern change. *Forensic science international*, 226(1-3):94–105, 2013.
- [40] Kai San Choi, Edmund Y Lam, and Kenneth KY Wong. Source camera identification using footprints from lens aberration. In *Digital Photography II*, volume 6069, pages 172–179. SPIE, 2006.
- [41] Vincent Christlein, Christian Riess, Johannes Jordan, Corinna Riess, and Elli Angelopoulou. An evaluation of popular copy-move forgery detection approaches. *IEEE Transactions on information forensics and security*, 7(6):1841–1854, 2012.
- [42] Kuo-Liang Chung, Wei-Jen Yang, Wen-Ming Yan, and Chung-Chou Wang. Demosaicing of color filter array captured images using gradient edge detection masks and adaptive heterogeneity-projection. *IEEE Transactions on Image Processing*, 17(12):2356–2367, 2008.
- [43] Gabriela Ciuperca, Andrea Ridolfi, and Jérôme Idier. Penalized maximum likelihood estimator for normal mixtures. *Scandinavian Journal of Statistics*, 30(1):45–59, 2003.
- [44] Aviad Cohen, Nir Nissim, and Yuval Elovici. Maljpeg: Machine learning based solution for the detection of malicious jpeg images. *IEEE Access*, 8:19997–20011, 2020.
- [45] Valentina Conotter, Pedro Comesana, and Fernando Pérez-González. Forensic detection of processing operator chains: Recovering the history of filtered jpeg images. *IEEE Transactions on Information Forensics and Security*, 10(11):2257–2269, 2015.
- [46] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212, 2004.
- [47] Ronald E Crochiere and Lawrence R Rabiner. *Multirate digital signal processing*. Prentice-Hall, 1983.
- [48] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.
- [49] Nahuel Dalgaard, Carlos Mosquera, and Fernando Pérez-González. On the role of differentiation for resampling detection. In *Proceedings of the 17th IEEE International Conference on Image Processing*, pages 1753–1756, 2010.

- [50] Neil E Day. Estimating the components of a mixture of normal distributions. *Biometrika*, 56(3):463–474, 1969.
- [51] Carl De Boor. *A Practical Guide to Splines*. Springer-Verlag, 1978.
- [52] Tiago José De Carvalho, Christian Riess, Elli Angelopoulou, Helio Pedrini, and Anderson de Rezende Rocha. Exposing digital image forgeries by illumination color classification. *IEEE Transactions on Information Forensics and Security*, 8(7):1182–1194, 2013.
- [53] Aaron Deever, Mrityunjay Kumar, and Bruce Pillman. Digital camera image formation: Processing and storage. In *Digital Image Forensics: There is More to a Picture than Meets the Eye*, pages 45–77. Springer New York, 2013.
- [54] Frédéric Devernay and Olivier D Faugeras. Automatic calibration and removal of distortion from scenes of structured environments. In *Investigative and Trial Image Processing*, volume 2567, pages 62–72. International Society for Optics and Photonics, 1995.
- [55] Ahmet Emir Dirik and Nasir Memon. Image tamper detection based on demosaicing artifacts. In *2009 16th IEEE International Conference on Image Processing*, pages 1497–1500, 2009.
- [56] Jimmie D Eggerton and Mandyam D Srinath. Statistical distributions of image dct coefficients. *Computers & electrical engineering*, 12(3-4):137–145, 1986.
- [57] Zhigang Fan and Ricardo L De Queiroz. Identification of bitmap compression history: Jpeg detection and quantizer estimation. *IEEE Transactions on Image Processing*, 12(2):230–235, 2003.
- [58] Hilda Faraji and W James MacLean. Ccd noise removal in digital images. *IEEE Transactions on Image Processing*, 15(9):2676–2685, 2006.
- [59] Hany Farid. Blind inverse gamma correction. *IEEE transactions on image processing*, 10(10):1428–1433, 2001.
- [60] Hany Farid. Digital doctoring: how to tell the real from the fake. *Significance*, 3(4):162–166, 2006.
- [61] Hany Farid. Image forgery detection. *IEEE Signal processing magazine*, 26(2):16–25, 2009.
- [62] Hany Farid. Exposing digital forgeries from jpeg ghosts. *IEEE Transactions on Information Forensics and Security*, 4(1):154–160, 2009.
- [63] Pasquale Ferrara, Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva. Image forgery localization via fine-grained analysis of cfa artifacts. *IEEE Transactions on Information Forensics and Security*, 7(5):1566–1577, 2012.
- [64] Anselmo Ferreira, Siovani C Felipussi, Carlos Alfaro, Pablo Fonseca, John E Vargas-Munoz, Jefersson A Dos Santos, and Anderson Rocha. Behavior knowledge space-based fusion for copy–move forgery detection. *IEEE Transactions on Image Processing*, 25(10):4729–4742, 2016.

- [65] William D Ferreira, Cristiane BR Ferreira, Gelson da Cruz Júnior, and Fabrizio Soares. A review of digital image forensics. *Computers & Electrical Engineering*, 85:106685, 2020.
- [66] Mario A. T. Figueiredo and Anil K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on pattern analysis and machine intelligence*, 24(3):381–396, 2002.
- [67] Alessandro Foi, Mejdí Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, 2008.
- [68] Jessica Fridrich, Miroslav Goljan, Dorin Hoge, and David Soukal. Quantitative steganalysis of digital images: estimating the secret message length. *Multimedia systems*, 9(3):288–302, 2003.
- [69] Jessica Fridrich, David Soukal, and Jan Lukáš. Detection of copy-move forgery in digital images. In *Proceedings of Digital Forensic Research Workshop*, pages 19–23, 2003.
- [70] Gary L Friedman. The trustworthy digital camera: Restoring credibility to the photographic image. *IEEE Transactions on consumer electronics*, 39(4):905–910, 1993.
- [71] Huazhu Fu and Xiaochun Cao. Forgery authentication in extreme wide-angle lens using distortion cue and fake saliency map. *IEEE Transactions on Information Forensics and Security*, 7(4):1301–1314, 2012.
- [72] Edward Furman and Ričardas Zitikas. A monotonicity property of the composition of regularized and inverted-regularized gamma functions with applications. *Journal of Mathematical Analysis and Applications*, 348(2):971–976, 2008.
- [73] Andrew C Gallagher. Detection of linear and cubic interpolation in jpeg compressed images. In *The 2nd Canadian Conference on Computer and Robot Vision*, pages 65–72, 2005.
- [74] Andrew C Gallagher and Tsuhan Chen. Image authentication by detecting traces of demosaicing. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, 2008.
- [75] Shang Gao, Guanshuo Xu, and Rui-Min Hu. Camera model identification based on the characteristic of cfa and interpolation. In *International Workshop on Digital Watermarking*, pages 268–280, 2011.
- [76] Pascal Getreuer. Linear methods for image interpolation. *Image Processing On Line*, 1:238–259, 2011.
- [77] Quentin Giboulot, Rémi Cogramme, and Patrick Bas. Detectability-based jpeg steganography modeling the processing pipeline: the noise-content trade-off. *IEEE Transactions on Information Forensics and Security*, 16:2202–2217, 2021.

- [78] Thomas Gloe and Rainer Böhme. The dresden image database for benchmarking digital image forensics. *Journal of Digital Forensic Practice*, 3(2-4):150–159, 2010.
- [79] Edgar González Fernández, Ana Sandoval Orozco, Luis García Villalba, and Julio Hernandez-Castro. Digital image tamper detection technique based on spectrum analysis of cfa artifacts. *Sensors*, 18(9):2804, 2018.
- [80] Atanas Gotchev, Karen Egiazarian, and Tapio Saramäki. Image interpolation by optimized spline-based kernels. *Advances in Signal transforms: Theory and Applications*, 7:285–335, 2007.
- [81] Izrail Solomonovich Gradshteyn and Iosif Moiseevich Ryzhik. *Table of Integrals, Series, and Products*. Elsevier, 7th edition, 2007.
- [82] Michael D Grossberg and Shree K Nayar. What can be known about the radiometric response from images? In *European conference on computer vision*, pages 189–205. Springer, 2002.
- [83] Christine Guillemot and Olivier Le Meur. Image inpainting: Overview and recent advances. *IEEE Signal Processing Magazine*, 31(1):127–144, 2013.
- [84] Bahadır K Gunturk, John Glotzbach, Yucel Altunbasak, Ronald W Schafer, and Russel M Mersereau. Demosaicking: color filter array interpolation. *IEEE Signal processing magazine*, 22(1):44–54, 2005.
- [85] Zhongwei He, Wei Lu, Wei Sun, and Jiwu Huang. Digital image splicing detection based on markov features in dct and dwt domain. *Pattern recognition*, 45(12):4292–4299, 2012.
- [86] Glenn E Healey and Raghava Kondepudy. Radiometric ccd camera calibration and noise estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):267–276, 1994.
- [87] Hagit Hel-Or and Ido Yerushalmy. Camera-based image forgery detection. In *Handbook of Digital Forensics of Multimedia Data and Devices*, pages 522–571. John Wiley & Sons, 2015.
- [88] Taras Holotyak, Jessica Fridrich, and David Soukal. Stochastic approach to secret message length estimation in $\pm k$ embedding steganography. In *Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 673–684. International Society for Optics and Photonics, 2005.
- [89] Yu-Feng Hsu and Shih-Fu Chang. Detecting image splicing using geometry invariants and camera characteristics consistency. In *2006 IEEE International Conference on Multimedia and Expo*, pages 549–552, 2006.
- [90] Yu-Feng Hsu and Shih-Fu Chang. Camera response functions for image forensics: an automatic algorithm for splicing detection. *IEEE Transactions on Information Forensics and Security*, 5(4):816–825, 2010.

- [91] Hailing Huang, Weiqiang Guo, and Yu Zhang. Detection of copy-move forgery in digital images using sift algorithm. In *IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, volume 2, pages 272–276, 2008.
- [92] Abir Jaafar Hussain, Ali Al-Fayadh, and Naeem Radi. Image compression techniques: A survey in lossless and lossy algorithms. *Neurocomputing*, 300:44–69, 2018.
- [93] Chryssanthi Iakovidou, Markos Zampoglou, Symeon Papadopoulos, and Yiannis Kompatsiaris. Content-aware detection of jpeg grid inconsistencies for intuitive image forensics. *Journal of Visual Communication and Image Representation*, 54:155–170, 2018.
- [94] Ankit Kumar Jaiswal and Rajeev Srivastava. Forensic image analysis using inconsistent noise pattern. *Pattern Analysis and Applications*, 24(2):655–667, 2021.
- [95] Jireh Jam, Connah Kendrick, Kevin Walker, Vincent Drouard, Jison Gee-Sern Hsu, and Moi Hoon Yap. A comprehensive review of past and present image inpainting methods. *Computer Vision and Image Understanding*, 203:103147, 2021.
- [96] Jong Ju Jeon, Hyun Jun Shin, and Il Kyu Eom. Estimation of bayer cfa pattern configuration based on singular value decomposition. *EURASIP Journal on Image and Video Processing*, 2017 (47):1–11, 2017.
- [97] Libin Jin, Wangli Xu, Liping Zhu, and Lixing Zhu. Penalized maximum likelihood estimator for skew normal mixtures. *arXiv preprint arXiv:1608.01513*, 2016.
- [98] Micah K Johnson and Hany Farid. Exposing digital forgeries through chromatic aberration. In *Proceedings of the 8th Workshop on Multimedia and Security*, pages 48–55, 2006.
- [99] Norman L Johnson, Samuel Kotz, and Narayanaswamy Balakrishnan. *Continuous Univariate Distributions, Volume 1*. John Wiley & Sons, 2 edition, 1994.
- [100] Jean-Michel Jolion. Images and benford’s law. *Journal of Mathematical Imaging and Vision*, 14 (1):73–81, 2001.
- [101] Pravin Kakar, Natarajan Sudha, and Wee Ser. Exposing digital image forgeries by detecting discrepancies in motion blur. *IEEE Transactions on Multimedia*, 13(3):443–452, 2011.
- [102] XiaoBing Kang and ShengMin Wei. Identifying tampered regions using singular value decomposition in digital image forensics. In *International Conference on Computer Science and Software Engineering*, volume 3, pages 926–930, 2008.
- [103] Robert Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160, 1981.

- [104] Jack Kiefer and Jacob Wolfowitz. Consistency of the maximum likelihood estimator in the presence of infinitely many incidental parameters. *The Annals of Mathematical Statistics*, pages 887–906, 1956.
- [105] Baek-Kyu Kim and Rae-Hong Park. Detection and correction of purple fringing using color desaturation in the xy chromaticity diagram and the gradient information. *Image and Vision Computing*, 28(6):952–964, 2010.
- [106] Matthias 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*, pages 11–20, 2008.
- [107] Matthias Kirchner. On the detectability of local resampling in digital images. In *Proceedings of SPIE 6819, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, page 68190F, 2008.
- [108] Matthias Kirchner. Linear row and column predictors for the analysis of resized images. In *Proceedings of the 12th ACM workshop on Multimedia and security*, pages 13–18, 2010.
- [109] Matthias Kirchner and Sujoy Chakraborty. A second look at first significant digit histogram restoration. In *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2015.
- [110] Matthias Kirchner and Thomas Gloe. On resampling detection in re-compressed images. In *Proceedings of the 1st IEEE International Workshop on Information Forensics and Security*, pages 21–25, 2009.
- [111] Keith Knight. *Mathematical Statistics*. Chapman & Hall/CRC, 2000.
- [112] Jan Kodovský and Jessica Fridrich. Calibration revisited. In *Proceedings of the 11th ACM workshop on Multimedia and security*, pages 63–74, 2009.
- [113] Jan Kodovský and Jessica Fridrich. Quantitative structural steganalysis of jsteg. *IEEE Transactions on Information Forensics and Security*, 5(4):681–693, 2010.
- [114] Paweł Korus. Digital image integrity—a survey of protection and verification techniques. *Digital Signal Processing*, 71:1–26, 2017.
- [115] Paweł Korus and Jiwu Huang. Multi-scale analysis strategies in prnu-based tampering localization. *IEEE Transactions on Information Forensics and Security*, 12(4):809–824, 2016.
- [116] Paweł Korus and Jiwu Huang. Multi-scale fusion for improved localization of malicious tampering in digital images. *IEEE Transactions on Image Processing*, 25(3):1312–1326, 2016.

- [117] Samuel Kotz, Tomasz Kozubowski, and Krzysztof Podgorski. *The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance*. Springer Science & Business Media, 2001.
- [118] Dirk P Kroese, Thomas Taimre, and Zdravko I Botev. *Handbook of Monte Carlo Methods*, volume 706 of *Wiley series in probability and statistics*. John Wiley & Sons, 2013.
- [119] Darwin T Kuan, Alexander A Sawchuk, Timothy C Strand, and Pierre Chavel. Adaptive noise smoothing filter for images with signal-dependent noise. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2):165–177, 1985.
- [120] Jeffrey C Lagarias, James A Reeds, Margaret H Wright, and Paul E Wright. Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on optimization*, 9(1):112–147, 1998.
- [121] Edmund Y Lam and George SK Fung. Automatic white balancing in digital photography. In *Single-Sensor Imaging: Methods and Applications for Digital Cameras*, pages 287–314. CRC Press, 2018.
- [122] Edmund Y Lam and Joseph W Goodman. A mathematical analysis of the dct coefficient distributions for images. *IEEE Transactions on Image Processing*, 9(10):1661–1666, 2000.
- [123] Eric Cooper Larson and Damon Michael Chandler. Most apparent distortion: full-reference image quality assessment and the role of strategy. *Journal of electronic imaging*, 19(1):011006, 2010.
- [124] Bhagwandas Pannalal Lathi and Roger A Green. *Essentials of digital signal processing*. Cambridge University Press, 2014.
- [125] Nhan Le and Florent ReTraint. Statistical detector of resampled tiff images. In *2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 398–401, 2018.
- [126] Nhan Le and Florent ReTraint. An improved algorithm for digital image authentication and forgery localization using demosaicing artifacts. *IEEE Access*, 7:125038–125053, 2019.
- [127] Nhan Le and Florent ReTraint. A recurrent framework for statistical modeling of jpeg ac dct coefficients. *Submitted to IEEE Access*, 2021.
- [128] Nhan Le and Florent ReTraint. Statistical modeling framework for ac dct coefficients of tampered jpeg images and application in forgery localization. *Submitted to IEEE Transactions on Information Forensics and Security*, 2021.
- [129] Erich L Lehmann and Joseph P Romano. *Testing statistical hypotheses*. Springer Science & Business Media, 2006.

- [130] Bin Li, Yun Q Shi, and Jiwu Huang. Detecting doubly compressed jpeg images by using mode based first digit features. In *10th IEEE Workshop on Multimedia Signal Processing*, pages 730–735, 2008.
- [131] Bin Li, Tian-Tsong Ng, Xiaolong Li, Shunquan Tan, and Jiwu Huang. Revealing the trace of high-quality jpeg compression through quantization noise analysis. *IEEE Transactions on Information Forensics and Security*, 10(3):558–573, 2015.
- [132] Bin Li, Tian-Tsong Ng, Xiaolong Li, Shunquan Tan, and Jiwu Huang. Statistical model of jpeg noises and its application in quantization step estimation. *IEEE Transactions on Image Processing*, 24(5):1471–1484, 2015.
- [133] Jixian Li, Wei Lu, Jian Weng, Yijun Mao, and Guoqiang Li. Double jpeg compression detection based on block statistics. *Multimedia Tools and Applications*, 77(24):31895–31910, 2018.
- [134] Lu Li, Jianru Xue, Xiaofeng Wang, and Lihua Tian. A robust approach to detect digital forgeries by exploring correlation patterns. *Pattern Analysis and Applications*, 18(2):351–365, 2015.
- [135] Weihai Li, Yuan Yuan, and Nenghai Yu. Passive detection of doctored jpeg image via block artifact grid extraction. *Signal Processing*, 89(9):1821–1829, 2009.
- [136] Xiang Lin, Jian-Hua Li, Shi-Lin Wang, Feng Cheng, Xiao-Sa Huang, et al. Recent advances in passive digital image security forensics: A brief review. *Engineering*, 4(1):29–39, 2018.
- [137] Zhouchen Lin, Rongrong Wang, Xiaoou Tang, and Heung-Yeung Shum. Detecting doctored images using camera response normality and consistency. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1087–1092, 2005.
- [138] Zhouchen Lin, Junfeng He, Xiaoou Tang, and Chi-Keung Tang. Fast, automatic and fine-grained tampered jpeg image detection via dct coefficient analysis. *Pattern Recognition*, 42(11):2492–2501, 2009.
- [139] Xianjin Liu, Wei Lu, Qin Zhang, Jiwu Huang, and Yun-Qing Shi. Downscaling factor estimation on pre-jpeg compressed images. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(3):618–631, 2020.
- [140] David G Lowe. Object recognition from local scale-invariant features. int. *Journal of Computer Vision*, 60(2):91–110, 2004.
- [141] Florian Luisier, Thierry Blu, and Michael Unser. Image denoising in mixed poisson–gaussian noise. *IEEE Transactions on image processing*, 20(3):696–708, 2010.
- [142] Rastislav Lukac. *Single-Sensor Imaging: Methods and Applications for Digital Cameras*. CRC Press, 2018.

- [143] Weiqi Luo, Jiwu Huang, and Guoping Qiu. Jpeg error analysis and its applications to digital image forensics. *IEEE Transactions on Information Forensics and Security*, 5(3):480–491, 2010.
- [144] Siwei Lyu, Xunyu Pan, and Xing Zhang. Exposing region splicing forgeries with blind local noise estimation. *International journal of computer vision*, 110(2):202–221, 2014.
- [145] Einar Maeland. On the comparison of interpolation methods. *IEEE transactions on medical imaging*, 7(3):213–217, 1988.
- [146] Babak Mahdian and Stanislav Saic. On periodic properties of interpolation and their application to image authentication. In *Proceedings of the 3rd International Symposium on Information Assurance and Security*, pages 439–446, 2007.
- [147] Babak Mahdian and Stanislav Saic. Blind authentication using periodic properties of interpolation. *IEEE Transactions on Information Forensics and Security*, 3(3):529–538, 2008.
- [148] Babak Mahdian and Stanislav Saic. Using noise inconsistencies for blind image forensics. *Image and Vision Computing*, 27(10):1497–1503, 2009.
- [149] Markku Makitalo and Alessandro Foi. Optimal inversion of the generalized anscombe transformation for poisson-gaussian noise. *IEEE Transactions on Image Processing*, 22(1):91–103, 2012.
- [150] Owen Mayer and Matthew C Stamm. Accurate and efficient image forgery detection using lateral chromatic aberration. *IEEE transactions on information forensics and security*, 13(7):1762–1777, 2018.
- [151] Geoffrey J McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*. Wiley series in probability and statistics. John Wiley & Sons, 2nd edition, 2007.
- [152] Geoffrey J McLachlan and David Peel. *Finite mixture models*. Wiley Series in Probability and Statistics. John Wiley & Sons, 2004.
- [153] Daniele Menon and Giancarlo Calvagno. Color image demosaicking: An overview. *Signal Processing: Image Communication*, 26(8-9):518–533, 2011.
- [154] M Kivanc Mihcak, Igor Kozintsev, and Kannan Ramchandran. Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 3253–3256, 1999.
- [155] Simone Milani, Marco Tagliasacchi, and Stefano Tubaro. Discriminating multiple jpeg compressions using first digit features. *APSIPA Transactions on Signal and Information Processing*, 3, 2014.
- [156] Tajuddin Manhar Mohammed, Jason Bunk, Lakshmanan Nataraj, Jawadul H Bappy, Arjuna Flenner, BS Manjunath, Shivkumar Chandrasekaran, Amit K Roy-Chowdhury, and Lawrence A Peterson. Boosting image forgery detection using resampling features and copy-move analysis. *Electronic Imaging*, 2018(7):118–1–7, 2018.

- [157] F Müller. Distribution shape of two-dimensional dct coefficients of natural images. *Electronics Letters*, 29(22):1935–1936, 1993.
- [158] Saralees Nadarajah. Gaussian dct coefficient models. *Acta applicandae mathematicae*, 106(3):455–472, 2009.
- [159] Tian-Tsong Ng, Shih-Fu Chang, and Mao-Pei Tsui. Using geometry invariants for camera response function estimation. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
- [160] Tina Nikoukhah, Miguel Colom, Jean-Michel Morel, and Rafael Grompone von Gioi. Local jpeg grid detector via blocking artifacts, a forgery detection tool. *Image Processing On Line*, 10:24–42, 2020.
- [161] Yakun Niu, Benedetta Tondi, Yao Zhao, and Mauro Barni. Primary quantization matrix estimation of double compressed jpeg images via cnn. *IEEE Signal Processing Letters*, 27:191–195, 2019.
- [162] Adriana Olmos and Frederick AA Kingdom. A biologically inspired algorithm for the recovery of shading and reflectance images. *Perception*, 33(12):1463–1473, 2004.
- [163] Xunyu Pan and Siwei Lyu. Region duplication detection using image feature matching. *IEEE Transactions on Information Forensics and Security*, 5(4):857–867, 2010.
- [164] J Anthony Parker, Robert V Kenyon, and Donald E Troxel. Comparison of interpolating methods for image resampling. *IEEE Transactions on medical imaging*, 2(1):31–39, 1983.
- [165] Cecilia Pasquini and Rainer Böhme. Information-theoretic bounds for the forensic detection of downsampled signals. *IEEE Transactions on Information Forensics and Security*, 14(7):1928–1943, 2019.
- [166] Cecilia Pasquini, Giulia Boato, and Fernando Pérez-González. Multiple jpeg compression detection by means of benford-fourier coefficients. In *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 113–118. IEEE, 2014.
- [167] Cecilia Pasquini, Giulia Boato, and Fernando Pérez-González. Statistical detection of jpeg traces in digital images in uncompressed formats. *IEEE Transactions on Information Forensics and Security*, 12(12):2890–2905, 2017.
- [168] Cecilia Pasquini, Irene Amerini, and Giulia Boato. Media forensics on social media platforms: a survey. *EURASIP Journal on Information Security*, 2021(1):1–19, 2021.
- [169] Anjie Peng, Yadong Wu, and Xiangui Kang. Revealing traces of image resampling and resampling antiforensics. *Advances in Multimedia*, 2017:ID 7130491, 13 pages, 2017.

- [170] Lionel Pibre, Jérôme Pasquet, Dino Ienco, and Marc Chaumont. Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover sourcemismatch. *Electronic Imaging*, 2016(8):1–11, 2016.
- [171] Alessandro Piva. An overview on image forensics. *International Scholarly Research Notices: Signal Processing*, 2013:ID 496701, 22 pages, 2013.
- [172] Christine I Podilchuk and Edward J Delp. Digital watermarking: algorithms and applications. *IEEE signal processing Magazine*, 18(4):33–46, 2001.
- [173] Alin C Popescu and Hany Farid. Exposing digital forgeries by detecting duplicated image regions. Technical Report TR 2004-515, Department of Computer Science, Dartmouth College, Hanover, 2004.
- [174] Alin C Popescu and Hany Farid. Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing*, 53(10):3948–3959, 2005.
- [175] Alin C Popescu and Hany Farid. Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on signal processing*, 53(2):758–767, 2005.
- [176] S Prasad and K R Ramakrishnan. On resampling detection and its application to detect image tampering. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 1325–1328, 2006.
- [177] William K Pratt. Digital image processing. 1978. *New York: Wiley*, 1978.
- [178] Tong Qiao, Florent Reiraint, Rémi Cogranne, and Cathel Zitzmann. Steganalysis of jsteg algorithm using hypothesis testing theory. *EURASIP Journal on Information Security*, 2015(1):1–16, 2015.
- [179] Tong Qiao, Florent Reiraint, Rémi Cogranne, and Thanh Hai Thai. Individual camera device identification from jpeg images. *Signal Processing: Image Communication*, 52:74–86, 2017.
- [180] Tong Qiao, Ran Shi, Xiangyang Luo, Ming Xu, Ning Zheng, and Yiming Wu. Statistical model-based detector via texture weight map: Application in re-sampling authentication. *IEEE Transactions on Multimedia*, 21(5):1077–1092, 2018.
- [181] Tong Qiao, Aichun Zhu, and Florent Reiraint. Exposing image resampling forgery by using linear parametric model. *Multimedia Tools and Applications*, 77(2):1501–1523, 2018.
- [182] Tong Qiao, Ran Shi, Xiangyang Luo, Ming Xu, Ning Zheng, and Yiming Wu. Statistical model-based detector via texture weight map: Application in re-sampling authentication. *IEEE Transactions on Multimedia*, 21(5):1077–1092, 2019.
- [183] Yong Song Qin and Bruce Smith. Likelihood ratio test for homogeneity in normal mixtures in the presence of a structural parameter. *Statistica Sinica*, pages 1165–1177, 2004.

- [184] Zhen Qin, Qingliang Zeng, Yixin Zong, and Fan Xu. Image inpainting based on deep learning: A review. *Displays*, 69:102028, 2021.
- [185] Rajeev Ramanath, Wesley E Snyder, Griff L Bilbro, and William A Sander. Demosaicking methods for bayer color arrays. *Journal of Electronic Imaging*, 11(3):306–315, 2002.
- [186] Rajeev Ramanath, Wesley E Snyder, Youngjun Yoo, and Mark S Drew. Color image processing pipeline. *IEEE Signal Processing Magazine*, 22(1):34–43, 2005.
- [187] Makkena Purnachandra Rao, AN Rajagopalan, and Guna Seetharaman. Harnessing motion blur to unveil splicing. *IEEE transactions on information forensics and security*, 9(4):583–595, 2014.
- [188] Randall Reiningger and Jerry Gibson. Distributions of the two-dimensional dct coefficients for images. *IEEE Transactions on Communications*, 31(6):835–839, 1983.
- [189] Florent Reiraint and Cathel Zitzmann. Quality factor estimation of jpeg images using a statistical model. *Digital Signal Processing*, 103:102759, 2020.
- [190] Peter J Rousseeuw and Christophe Croux. Alternatives to the median absolute deviation. *Journal of the American Statistical association*, 88(424):1273–1283, 1993.
- [191] Seung-Jin Ryu, Min-Jeong Lee, and Heung-Kyu Lee. Detection of copy-rotate-move forgery using zernike moments. In *International workshop on information hiding*, pages 51–65, 2010.
- [192] Husrev Taha Sencar and Nasir Memon. *Digital Image Forensics: There is More to a Picture than Meets the Eye*. Springer New York, 2013.
- [193] Hyun Jun Shin, Jong Ju Jeon, and Il Kyu Eom. Color filter array pattern identification using variance of color difference image. *Journal of Electronic Imaging*, 26(4):043015, 2017.
- [194] Amneet Singh, Gurinder Singh, and Kulbir Singh. A markov based image forgery detection approach by analyzing cfa artifacts. *Multimedia Tools and Applications*, 77(21):28949–28968, 2018.
- [195] Raahat Devender Singh and Naveen Aggarwal. Detection and localization of copy-paste forgeries in digital videos. *Forensic science international*, 281:75–91, 2017.
- [196] Brian C Smith and Lawrence A Rowe. Algorithms for manipulating compressed images. *IEEE Computer Graphics and Applications*, 13(5):34–42, 1993.
- [197] Geun-Sil Song, Yong-In Yun, and Won-Hyung Lee. A new estimation approach of resampling factors using threshold-based peak detection. In *Proceedings of the IEEE International Conference on Consumer Electronics*, pages 731–732, 2011.
- [198] Matthew C Stamm and KJ Ray Liu. Anti-forensics of digital image compression. *IEEE Transactions on Information Forensics and Security*, 6(3):1050–1065, 2011.

- [199] Ashwin Swaminathan, Min Wu, and KJ Ray Liu. Nonintrusive component forensics of visual sensors using output images. *IEEE Transactions on Information Forensics and Security*, 2(1):91–106, 2007.
- [200] Ashwin Swaminathan, Min Wu, and KJ Ray Liu. Digital image forensics via intrinsic fingerprints. *IEEE transactions on information forensics and security*, 3(1):101–117, 2008.
- [201] Théo Taburet, Patrick Bas, Wadih Sawaya, and Jessica Fridrich. Natural steganography in jpeg domain with a linear development pipeline. *IEEE Transactions on Information Forensics and Security*, 16:173–186, 2020.
- [202] Jun Takamatsu, Yasuyuki Matsushita, Tsukasa Ogasawara, and Katsushi Ikeuchi. Estimating demosaicing algorithms using image noise variance. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 279–286, 2010.
- [203] Songpon Teerakanok and Tetsutaro Uehara. Copy-move forgery detection: A state-of-the-art technical review and analysis. *IEEE Access*, 7:40550–40568, 2019.
- [204] Thanh Hai Thai and Rémi Cogramne. Estimation of primary quantization steps in double-compressed jpeg images using a statistical model of discrete cosine transform. *IEEE Access*, 7:76203–76216, 2019.
- [205] Thanh Hai Thai, Remi Cogramne, and Florent Reiraint. Camera model identification based on the heteroscedastic noise model. *IEEE Transactions on Image Processing*, 23(1):250–263, 2013.
- [206] Thanh Hai Thai, Remi Cogramne, and Florent Reiraint. Statistical model of quantized dct coefficients: Application in the steganalysis of jsteg algorithm. *IEEE Transactions on Image Processing*, 23(5):1980–1993, 2014.
- [207] Thanh Hai Thai, Florent Reiraint, and Rémi Cogramne. Generalized signal-dependent noise model and parameter estimation for natural images. *Signal Processing*, 114:164–170, 2015.
- [208] Thanh Hai Thai, Florent Reiraint, and Rémi Cogramne. Camera model identification based on the generalized noise model in natural images. *Digital Signal Processing*, 48:285–297, 2016.
- [209] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. Deepfakes and beyond: A survey of face manipulation and fake detection. *Information Fusion*, 64:131–148, 2020.
- [210] Chi-Yi Tsai and Kai-Tai Song. A new edge-adaptive demosaicing algorithm for color filter arrays. *Image and Vision Computing*, 25(9):1495–1508, 2007.
- [211] Chengjie Tu and Trac D Tran. Context-based entropy coding of block transform coefficients for image compression. *IEEE Transactions on Image Processing*, 11(11):1271–1283, 2002.

- [212] Luisa Verdoliva. Media forensics and deepfakes: an overview. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):910–932, 2020.
- [213] Gregory K Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.
- [214] Anding Wang, Xufen Xie, Hongyuan Wang, Nianyu Zou, and Yingying Shang. Precision-based regularization comparametric calibration method for imaging system response functions. *Infrared Physics & Technology*, 92:409–416, 2018.
- [215] Haixia Wang, Li Jiang, Ronghua Liang, and Xiao-Xin Li. Exemplar-based image inpainting using structure consistent patch matching. *Neurocomputing*, 269:90–96, 2017.
- [216] Jinwei Wang, Hao Wang, Jian Li, Xiangyang Luo, Yun-Qing Shi, and Sunil Kumar Jha. Detecting double jpeg compressed color images with the same quantization matrix in spherical coordinates. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(8):2736–2749, 2019.
- [217] Wei Wang, Jing Dong, and Tieniu Tan. Exploring dct coefficient quantization effects for local tampering detection. *IEEE Transactions on Information Forensics and Security*, 9(10):1653–1666, 2014.
- [218] Weihong Wang and Hany Farid. Exposing digital forgeries in video by detecting double mpeg compression. In *Proceedings of the 8th workshop on Multimedia and security*, pages 37–47, 2006.
- [219] Yubin Wu, Shixiong Jiang, Zhenkun Xu, Song Zhu, and Danhua Cao. Lens distortion correction based on one chessboard pattern image. *Frontiers of Optoelectronics*, 8(3):319–328, 2015.
- [220] Fei Xue, Ziyi Ye, Wei Lu, Hongmei Liu, and Bin Li. Mse period based estimation of first quantization step in double compressed jpeg images. *Signal Processing: Image Communication*, 57:76–83, 2017.
- [221] Fei Xue, Wei Lu, Ziyi Ye, and Hongmei Liu. Jpeg image tampering localization based on normalized gray level co-occurrence matrix. *Multimedia Tools and Applications*, 78(8):9895–9918, 2019.
- [222] En-Hui Yang, Xiang Yu, Jin Meng, and Chang Sun. Transparent composite model for dct coefficients: Design and analysis. *IEEE transactions on image processing*, 23(3):1303–1316, 2014.
- [223] Heng Yao, Shuozhong Wang, Xinpeng Zhang, Chuan Qin, and Jinwei Wang. Detecting image splicing based on noise level inconsistency. *Multimedia Tools and Applications*, 76(10):12457–12479, 2017.
- [224] Bee Wah Yap and Chiaw Hock Sim. Comparisons of various types of normality tests. *Journal of Statistical Computation and Simulation*, 81(12):2141–2155, 2011.

- [225] Ido Yerushalmy and Hagit Hel-Or. Digital image forgery detection based on lens and sensor aberration. *International Journal of Computer Vision*, 92(1):71–91, 2011.
- [226] Liyang Yu, Qi Han, Xiamu Niu, SM Yiu, Junbin Fang, and Ye Zhang. An improved parameter estimation scheme for image modification detection based on dct coefficient analysis. *Forensic science international*, 259:200–209, 2016.
- [227] Fabian Zach, Christian Riess, and Elli Angelopoulou. Automated image forgery detection through classification of jpeg ghosts. In *Joint DAGM and OAGM Symposium*, pages 185–194. Springer, 2012.
- [228] Ofer Zeitouni, Jacob Ziv, and Neri Merhav. When is the generalized likelihood ratio test optimal? *IEEE Transactions on Information Theory*, 38(5):1597–1602, 1992.
- [229] Jing Zhang, Zhanlei Feng, and Yuting Su. A new approach for detecting copy-move forgery in digital images. In *11th IEEE Singapore International Conference on Communication Systems*, pages 362–366, 2008.
- [230] Lei Zhang, Xiaolin Wu, Antoni Buades, and Xin Li. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *Journal of Electronic imaging*, 20(2):023016, 2011.
- [231] Rong Zhang and Rang-Ding Wang. In-camera jpeg compression detection for doubly compressed images. *Multimedia Tools and Applications*, 74(15):5557–5575, 2015.
- [232] Lilei Zheng, Ying Zhang, and Vrizlynn LL Thing. A survey on image tampering and its detection in real-world photos. *Journal of Visual Communication and Image Representation*, 58:380–399, 2019.
- [233] Nan Zhu, Junge Shen, and Xiaotong Niu. Double jpeg compression detection based on noise-free dct coefficients mixture histogram model. *Symmetry*, 11(9):1119, 2019.

Thi Ai Nhàn LÊ

Doctorat : Optimisation et Sûreté des Systèmes

Année 2021

Modélisation statistique pour la détection de la falsification des images numériques

À l'ère du numérique, la crédibilité du contenu des images est une préoccupation majeure en raison de la popularité des outils d'édition faciles à utiliser et peu coûteux. Les images falsifiées peuvent être utilisées pour manipuler l'opinion publique lors des élections, commettre des fraudes et discréditer ou faire chanter des personnes. Face à cette situation préoccupante, nous développons dans cette thèse trois techniques efficaces basées sur (i) les traces de dématricage (ii) les traces de compression JPEG, et (iii) les traces de rééchantillonnage pour détecter les images falsifiées et localiser les différents types de falsification. Bien que ces techniques soient différentes, elles fonctionnent sous l'hypothèse commune que les manipulations peuvent altérer certaines propriétés statistiques sous-jacentes des images naturelles. Un processus de détection en deux étapes a été adopté pour chaque technique de détection : (i) analyser et modéliser les caractéristiques statistiques des images authentiques et falsifiées, puis (ii) concevoir un détecteur statistique pour différencier les images falsifiées des images authentiques en estimant les changements dans leurs modèles. Diverses expérimentations numériques sur plusieurs ensembles de données de référence bien connus mettent en évidence la qualité des performances et la robustesse des techniques de détection proposées.

Mots clés : criminalistique – traitement d'images, techniques numériques – méthodes statistiques – modèles mathématiques – estimation de paramètres – tests d'hypothèses (statistique) – algorithmes EM.

Statistical Modeling for Detection of Digital Image Forgery

In today's digital age, the trustworthiness of image content is of great concern due to the dissemination of easy-to-use and low-cost image editing tools. Forged images can be used to manipulate public opinion during elections, commit fraud, discredit or blackmail people. Faced with such a serious situation, we develop in this doctoral project three versatile techniques based on (i) demosaicing traces (ii) JPEG compression traces, and (iii) resampling traces for detecting forged digital images and localizing various types of tampering therein. Although these techniques are different, they work under the common assumption that manipulations may alter some underlying statistical properties of natural images. A two-steps detection process has been adopted for every detection technique: (i) analyze and model statistical features of both the authentic and forged images associated with specific in-camera and/or post-camera traces, then (ii) design a statistical detector to differentiate between the authentic and forged images by estimating statistical changes in their models. Various numerical experiments on several well-known benchmark datasets highlight the performances and robustness of the proposed detection techniques.

Keywords: forensic sciences – image processing, digital techniques – statistical methods – mathematical models – parameter estimation – statistical hypothesis testing – expectation-maximization algorithms.

Thèse réalisée en partenariat entre :

