ABSTRACT

IMAGE STATISTICAL FRAMEWORKS FOR DIGITAL IMAGE FORENSICS

by
Patchara Sutthiwan

The advances of digital cameras, scanners, printers, image editing tools, smartphones, tablet personal computers as well as high-speed networks have made a digital image a conventional medium for visual information. Creation, duplication, distribution, or tampering of such a medium can be easily done, which calls for the necessity to be able to trace back the authenticity or history of the medium. Digital image forensics is an emerging research area that aims to resolve the imposed problem and has grown in popularity over the past decade. On the other hand, anti-forensics has emerged over the past few years as a relatively new branch of research, aiming at revealing the weakness of the forensic technology.

These two sides of research move digital image forensic technologies to the next higher level. Three major contributions are presented in this dissertation as follows.

First, an effective multi-resolution image statistical framework for digital image forensics of passive-blind nature is presented in the frequency domain. The image statistical framework is generated by applying Markovian rake transform to image luminance component. Markovian rake transform is the applications of Markov process to difference arrays which are derived from the quantized block discrete cosine transform 2-D arrays with multiple block sizes. The efficacy and universality of the framework is then evaluated in two major applications of digital image forensics: 1) digital image tampering detection; 2) classification of computer graphics and photographic images.

Second, a simple yet effective anti-forensic scheme is proposed, capable of obfuscating double JPEG compression artifacts, which may vital information for image forensics, for instance, digital image tampering detection. Shrink-and-zoom (SAZ) attack, the proposed scheme, is simply based on image resizing and bilinear interpolation. The effectiveness of SAZ has been evaluated over two promising double JPEG compression schemes and the outcome reveals that the proposed scheme is effective, especially in the cases that the first quality factor is lower than the second quality factor.

Third, an advanced textural image statistical framework in the spatial domain is proposed, utilizing local binary pattern (LBP) schemes to model local image statistics on various kinds of residual images including higher-order ones. The proposed framework can be implemented either in single- or multi-resolution setting depending on the nature of application of interest. The efficacy of the proposed framework is evaluated on two forensic applications: 1) steganalysis with emphasis on HUGO (Highly Undetectable Steganography), an advanced steganographic scheme embedding hidden data in a content-adaptive manner locally into some image regions which are difficult for modeling image statics; 2) image recapture detection (IRD). The outcomes of the evaluations suggest that the proposed framework is effective, not only for detecting local changes which is in line with the nature of HUGO, but also for detecting global difference (the nature of IRD).

IMAGE STATISTICAL FRAMEWORKS FOR DIGITAL IMAGE FORENSICS

by
Patchara Sutthiwan

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering

Department of Electrical and Computer Engineering

May 2012

.

APPROVAL PAGE

IMAGE STATISTICAL FRAMEWORKS FOR DIGITAL IMAGE FORENSICS

Patchara Sutthiwan

Dr. Yun Q. Shi, Dissertation Advisor Professor of Electrical and Computer Engineering, NJIT	Date
Dr. Atam P. Dhawan, Committee Member Distinguished Professor of Electrical and Computer Engineering, NJIT	Date
Dr. Mengchu Zhou, Committee Member Professor of Electrical and Computer Engineering, NJIT	Date
Dr. Roberto Rojas-Cessa, Committee Member Associate Professor of Electrical and Computer Engineering, NJIT	Date
Dr. Hong Zhao, Committee Member Assistant Professor of Computer Sciences and Engineering, FDU	Date

BIOGRAPHICAL SKETCH

Author: Patchara Sutthiwan

Degree: Doctor of Philosophy

Date: May 2012

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2012
- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2006
- Bachelor of Engineering in Electrical Engineering,
  Chulalongkorn University, Bangkok, Thailand, 2001

Major: Electrical Engineering

Presentations and Publications:

Y.Q. Shi, P. Sutthiwan, and L. Chen, "Textural features for steganalysis," *14th International Conference on Information Hiding*, Berkeley, California, May 2012 (accepted).

P. Sutthiwan, and Y.Q. Shi, "Anti-forensics of double JPEG compression detection," *10th International Workshop on Digital-Forensics and Watermarking*, pp. 411-424, Atlantic City, New Jersey, October 2011.

P. Sutthiwan, Y.Q. Shi, H. Zhao, T.T. Ng, and W. Su, "Markovian rake transform for digital image tampering detection," *Transactions on Data Hiding and Multimedia Security VI*, pp. 1-17, 2011.

P. Sutthiwan, Y.Q. Shi, W. Su, and T.T. Ng, "Rake transform and edge statistics for image forgery detection," *IEEE International Conference on Multimedia and Expo*, pp. 1463-1468, Singapore, July 2010.

P. Sutthiwan, Y.Q. Shi, J. Dong, T. Tan, and T.T. Ng, "New developments in color image tampering detection," *IEEE International Symposium on Circuits and Systems*, pp. 3064-3067, Paris, France, May 30-June 2 2010.

P. Sutthiwan, X. Cai, Y.Q. Shi, and H. Zhang, "Computer graphics classification based on Markov process model and boosting feature selection technique," *16th IEEE International Conference on Image Processing,* pp. 2913-2916, Cairo, Eqypt, November 2009.

P. Sutthiwan, J. Ye, and Y.Q. Shi, "An enhanced statistical approach to identifying photorealistic images," *8th International Workshop on Digital Watermarking*, pp. 323-335, Guildford, United Kingdom, August 2009.

A. Abdi, G. Huaihai, and P. Sutthiwan, "A new vector sensor receiver for underwater acoustic communication," *OCEANS*, pp. 1-10, Vancouver, British Columbia, Canada, September 29-October 4 2007.

To my beloved family

for their love, encouragement, faith, and support

which are the reasons for who I am and what I will be

ACKNOWLEDGMENT

This dissertation is the fruit of a number of people who have sharpened my insight and perception of reality. First and foremost, I owe my deepest gratitude to Dr. Yun Q. Shi, my dissertation advisor, for giving me the wondrous opportunity to be involved in frontier research, for advising me with kindness and patience, and for helping me to realize my perpetual insufficiency.

I am grateful to all dissertation committee members, Dr. Atam P. Dhawan, Dr. Mengchu Zhou, Dr. Roberto Rojas-Cessa, and Dr. Hong Zhao, for devoting their time and for constructive comments which strengthened this dissertation.

I am grateful to Dr. Guorong Xuan, Dr. Tian-Tsong Ng, and Dr. Jessica Fridrich for meaningful technical discussions which eventually led me to a better understanding of the pattern recognition theory. I am also grateful to my previous research advisors, Dr. Ali Abdi and Dr. Watcharapong Khovidhunkij, for preparing me with critical thinking and research skills.

I am grateful to Dr. Osvaldo Simeone, Dr. Ali N. Akansu, Dr. Richard A. Haddad, Dr. Alexander M. Haimovich, and Dr. Roy You, for inspiring me and keeping my academic interests alive. I am also grateful to all my teachers, whose names can never be completely listed on this acknowledgement page, for building my solid foundation in science and engineering.

I am grateful to all my colleagues, especially Haifeng Xiao, Guanshou Xu, Jingyu Ye, Fei Long, Shang Gao, Peng Meng, Nikhil Dubbaka, Dr. Chunhua Chen, Dr. Wen Chen, Dr. Bin Li, and Dr. Fangjun Huang, for their supports either in research or in my

personal life. I am also grateful to Nonjaros Chomcharn, Dr. Atipol Kanchanapiboon, and Dr. Sun Olapiriyakul, for being by my side through many ups and downs.

I am grateful to all faculty and staff of the Department of Electrical and Computer Engineering at New Jersey Institute of Technology, especially Dr. Nirwan Ansari, Dr. Durgamadhab Misra, Dr. Kenneth S. Sohn, Joan M. Mahon, Barbara Faltz, Tanita S. Evan, Maureen B. Fagan, and Jacinta Y. Williams, and Richard Mondalto, for their generosities and supports. I am also grateful to Clarisa Gonzalez-Lenahan and Dr. Marino Xanthos for proof-reading this dissertation and to the staff of the Office of Graduate Studies as well as the Office of International Students at New Jersey Institute of Technology for their kind supports and advices.

Last, but not least, my heartfelt gratitude belongs to my parents, siblings, and relatives, for their love, encouragement, faith, and support. Without them, I could never have come this far.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF TABLES
(Continued)

LIST OF FIGURES

LIST OF FIGURES
(Continued)

LIST OF SYMBOLS

©	Copyright
CG	Computer Graphics
PG	Photographic Images
JPEG	Joint Photographic Experts Group
DQ	Double Quantization
HUGO	Highly Undetectable Steganography
IRD	Image Recapture Detection
MRT	Markovian Rake Transform
SAZ	Shrink-and-Zoom Attack
LBP	Local Binary Pattern
SVM	Support Vector Machines
RBF	Radial Basis Function
FLD	Fisher's Linear Discriminant
%	Percentage Sign
TP	True Positive
TN	True Negative
AC	Accuracy
ROC	Receiver Operating Characteristic
AUC	Area Under the Operating Curve
PCA	Principle Component Analysis

CHAPTER 1

INTRODUCTION

This chapter presents the problem statement, background and the outline of this doctoral research. Firstly, the motivation and objectives are introduced. Secondly, passive-blind detection framework is discussed. Thirdly, the contributions of this dissertation are summarized. Finally, the outline of this dissertation is presented.

1.1 Motivation and Objectives

The capability that one can generate and manipulate a digital image as well as pass its visual information on to others via any kind of networks has been drastically increased, obviously bringing about many benefits to humankind yet potentially causing harm if ever such information is distorted. Therefore, the interest in and the necessity of claiming authenticity and originality of digital image have increased remarkably which led to the emergence of the research area of digital image forensics over the past decade.

According to the Air Force Research Laboratory [1], "digital data forensics refers to analysis that gathers evidence of data composition, origin, pedigree, etc." Digital image forensics, a branch of digital data forensics, is a broad research area focusing on the investigation of the authenticity, origin, and processing history of digital images, which spans a variety of related applications such as image steganalysis, image tampering detection, classification of computer graphics and photographic images, camera identification, double JPEG detection, image recapture detection and so on.

As sophisticated as digital image forensic schemes have been designed, they usually exhibit some shortcomings. That is, the performance of such schemes would drop generally seriously under some attack or circumstances. Digital image anti-forensics is a relatively new branch of research aiming at revealing the drawback of existing forensic schemes. The knowledge gained by this study is fruitful to the next generation of digital image forensic schemes. In other words, digital image anti-forensics is the key to a more robust forensic scheme which can potentially be rendered real-world applications.

In what follows, the brief introductions to a few chosen forensics application are to be discussed.

1.1.1 Image Tampering Detection

Digital visual information in the form of digital images and videos is becoming popular and important as broadband infrastructure and digital technology are growing. As such, how individuals perceive visual media is of importance and could play an important role in shaping their society; as a result, the need for digital image tampering detection has been arisen. There have been many real-life examples of the usage of image tampering to mislead society. Figure 1.1 depicts a recent example of it.

Figure 1.1 Example of real-life tampered image. (left) Prokhorov next to Navalny in the authentic image; (right) Navalny next to Berezovsky in the tampered image (adapted from [2]).

The Guardian reported on 9 January 2012 [2] that a faked picture of Aleksei Navalny, an activist, next to Boris A. Berezovsky, an exiled financier, was distributed in a Russian newspaper. The driving force behind the creation of this tampered image, as shown in Figure 1.1 (right), was to defame political enemies.

There is a wide variety of methods to forge digital images, the simplest among which could be a naïve cut-and-paste operation, also known as image splicing, simply taking a portion of an image and puts it onto either the same or another image. Whenever any post-processing such as in [3], [4] is further applied to image splicing, such a digital image forgery operation is generally defined by the term "image tampering."

Image tampering detection is undoubtedly more challenging than image splicing detection, since those post-processing techniques make image tampering much less perceptible to human eyes. However, the trace left behind by the editing remains detectable by computer. Therefore, the objective of image tampering detection is to distinguish between tampered and authentic images.

1.1.2 Classification of Computer Graphics and Photographic Images

Computer graphics (CG) have become more and more photorealistic due to the advancement made in rendering software. As a result, it has become very much difficult for people to visually differentiate them from photographic images (PG).

According to [5], in April 2002, the US Supreme Court overthrew the 1996 Child Pornography Prevention Act (CPPA) extending the existing federal criminal laws against child pornography to include certain types of "virtual porn." That is, "virtual" or "computer-generated" minor fictitious child porn images are constitutionally protected.

High photorealistic CG may be used as either a convincing form of image forgery or a replacement of hard-oriented scene in movie production; consequently, classifying CG from PG appears to be an important task in providing court evidence for child pornography, image forgery detection, and benchmarking the rendering software.

1.1.3 Double JPEG Compression Detection

JPEG (Joint Photographic Experts Group) compression provides a means to store digital images using less physical storage than it would have been required without data compression. Presently, saving digital images in JPEG format is a common practice.

When an uncompressed image is JPEG compressed, the resultant compressed image is technically called singly compressed; however, the situation in which a singly compressed image has been compressed one more time happens frequently. This calls for an effective means of distinguishing between singly and doubly compressed images (double JPEG compression detection) which reveals image processing history (one of the goals of digital image forensics).

1.1.4 Image Steganalysis

More than a decade, data hiding has become an important research area. Information can be hidden into digital media, for instance, audio signals, images, as well as videos, either noticeably or unnoticeably to human perception. The application of data hiding can mainly be categorized into digital watermarking and steganography.

Two main applications of digital watermarking are for copy right protection and authentication while that of steganography is for covert communication which requires

that the cover media embedded with secret messages be as similar to the original cover media as possible.

USA Today reported on 5 February 2001 [6] that Osama bin Laden and his associates were using steganography to hide essential information of terrorist plots on sport chat rooms as pornographic pictures on many web sites. This is one of the compelling forces behind the increasing popularity of steganalysis.

Image steganalysis is the art and science of detecting secret messages embedded by using any steganography. Image steganalysis can be considered as a branch of digital image forensics, which has relatively longer history. Most of early steganographic schemes globally embed information throughout a cover image, but these days, more advanced steganographic schemes such as HUGO (Highly Undetectable Steganography) [7] are more powerful in the sense that they tend to hide information in a given cover image locally, in particular, in some regions of the cover image which are difficult to model statistical changes. Therefore, early steganalysis schemes fail to detect stego images generated by recent steganographic schemes. Such state-of-the-art steganography then calls for more advanced steganalysis schemes.

1.1.5 Image Recapture Detection

The security system of next generation mobile device, ATM machine, and so on, may be enhanced by incorporating face verification system. Some companies, for instance [8, 9], have already incorporated face verification system into their products. It is not difficult to foresee that such a security system could be easily fooled around by faked identity which can be of a form of printed face images of authorized person. It is also possible to use recaptured images to mislead image tampering detection scheme to classify tampered

images as authentic images. Consequently, image recapture detection (IRD) schemes, which differentiate real-scene images from recaptured ones, have been called for.

1.2 Passive-Blind Detection

Detection methodologies in digital image forensics can be mainly bifurcated as follows: 1) active [10] detection; 2) passive [11] detection. Active detection detects the authenticity of digital watermark embedded before the reception of images. The greatest constraint of the implementation of this method is that it requires built-in standardized watermarking functionality in image capturing devices which has not yet become a reality.

Contrary to active detection, passive detection is unfettered by any required prior knowledge of images, solely exploiting the knowledge of images themselves. In digital image forensics, passive detection is without doubt overshadowing active detection and is only the detection methodology in this doctoral dissertation.

The key mechanism of digital image forensics under passive-blind framework in this dissertation is a combined usage of image statistical features and machine learning. Represented in the form of vector of length relatively much smaller than the total number of image pixels, image statistical features compactly represent the underlying image properties. Effective image statistical features should be able to statistically reveal the fundamental differences between different classes of images, and the effectiveness of image statistical features can be evaluated by any classifier such as Fisher's linear discriminant (FLD), support vector machines (SVM), and so on. That is, the emphasis of this doctoral research is to find effective image statistical features rather than to develop

classifiers. The forensic applications presented throughout this doctoral research work are formulated as binary classification problems where instances of interest are bifurcated into two different classes: positive and negative.

In this dissertation, the performance of a classification system is determined by true positive rate (TP rate), true negative rate (TN rate) and accuracy (AC). TP rate is the percentage of the positive instances correctly classified, TN rate being the percentage of negative instances correctly identified. AC is percentage of all the instances correctly classified.

1.3 Contribution of This Dissertation

This dissertation carries out three major contributions as follows:

1) An efficient multi-resolution image statistical framework in the frequency domain based on Markovian rake transform (MRT). The efficacy of MRT framework has been evaluated on digital image tampering detection as well as classification of computer graphics and photographic images.

2) An effective anti-forensic scheme capable of obfuscating double JPEG compression. Shrink-and-zoom (SAZ) attack, the proposed scheme, is the application of image resizing and bilinear interpolation. The efficacy of SAZ has been evaluated over two promising double JPEG compression schemes.

3) A novel textural image statistical framework in the spatial domain. The proposed framework employs local binary pattern (LBP) operators to model image statistics on a variety of residual images. The implementation of the proposed framework allows both single- or multi-resolution approaches. The efficacy of the proposed

framework has been evaluated on steganalysis with emphasis on HUGO (Highly Undetectable Steganography) and image recapture detection (IRD).

1.4 Dissertation Outline

Chapter 2 introduces the MRT framework and its application on digital image tampering detection on a public large-scale dataset as well as details the required dataset adjustment procedure.

Chapter 3 presents a scheme classifying computer graphics from photographic images based on compressed-domain techniques. The first technique utilizes Markovian features derived from JPEG coefficients of multiple color channels. The color channel selection is done according to correlation analysis. In the second technique, image features are generated by applying boosting feature selection on the features obtained using the first technique. The third technique is based on MRT framework.

Chapter 4 introduces SAZ attack, an anti-forensic technique based on the application of image re-sizing and bilinear interpolation, to suppress double JPEG compression artifacts and mislead two prominent double JPEG compression detection schemes.

Chapter 5 introduces textural image statistical framework, which employs LBP to model image statistics on a bunch of residual images, to solve steganalysis problem based entirely on HUGO. Many residual images discussed in this chapter are considered high-order and are generated using some textural image techniques such as Laws' mask, Markov Random Field neighbor, and clique.

Chapter 6 applies a small portion of textural image statistical framework intensively discussed in Chapter 5 to IRD application.

Chapter 7 gives a summary of this dissertation and points out to some potential future research work.

CHAPTER 2

IMAGE TAMPERING DETECTION

An effective framework for passive-blind color image tampering detection is presented. The proposed image statistical features are generated by applying Markovian rake transform to image luminance component. Markovian rake transform is the applications of Markov process to difference arrays which are derived from the quantized block discrete cosine transform 2-D arrays with multiple block sizes. The efficacy of thus generated features has been confirmed over a recently established large-scale image dataset designed for tampering detection, with which some relevant issues have been addressed and corresponding adjustment measures have been taken. The initial tests by using thus generated classifiers on some real-life tampered images available in the Internet show signs of promise of the proposed features as well as the challenge encountered by the research community of image tampering detection. The related works were published in [12, 13].

2.1 Introduction

An old saying, "Don't judge a book by its cover," has its root dating back before the Digital Revolution. Undeniably, however, image tampering is emphasizing the importance of this everlasting adage in modern society as splicing is a common way to distort semantic content of an image, which could lead the public into misbelieving the veracity behind the scene.

In this research work, image tampering is defined as any intentional alteration to a digital image in order to change its semantic meaning, i.e., malicious attacks. A cut-and-paste technique is fundamental to image tampering in both printed and digital media.

Such a technique takes a portion of an image and puts it onto either the same or another image. A cut-and-paste operation, also known as image splicing, is still the modus operandi in image tampering in digital images. However, the naive nature of image splicing often leaves noticeable trace on spliced images usually in terms of sharp boundaries of changed regions in such images owing to abrupt changes in pixel values.

Advancement in digital image processing has put image tampering into a more challenging stage. Less perceptible tampered images have been made possible by applying some image processing to spliced images, e.g., with some post-processing further applied to spliced image, such an operation is generally less identifiable to human beings than image splicing. Often, image tampering artifacts are so imperceptible that human eyes can hardly capture them; therefore, it is more reliable to detect tampered images using a computing machine.

A number of passive image tampering detection methods have been proposed over the past several years. Ng et al. [14] analyzed the presence of the abrupt changes of pixel values in spliced image and utilized higher order statistics along with bicoherence of images to form image features. Johnson and Farid [15, 16] proposed a method to detecting image tampering which relies primarily on the lighting inconsistency in an image. The downside to their method is that it is not capable of detecting tampered images created by two images taken under virtually the same or rather similar light conditions. Hsu and Chang [17] employed geometric invariants and camera characteristic

consistency to interactively detect spliced images. Fu et al. [18] constructed image features for image splicing detection from Hilbert-Huang transform and moment of characteristic function of wavelet sub-bands. In Chen et al. [19], image features are derived from 2-D phase congruency and statistical moments of characteristic function. The natural image model in [20] consists of statistical features extracted from a test image and its corresponding 2-D arrays generated by applying multi-size block discrete cosine transform (MBDCT) to the test image. The statistical features are formulated by the combination of moments of characteristic functions of wavelet sub-bands and Markov transition probabilities of difference 2-D arrays. Dong et al. [21] considered the discontinuity of image pixel correlation and coherency caused by splicing as keys observation. The statistical features [21] are extracted from image run-length representation and image edge statistics. Farid [22] proposed a scheme to detect tampering based on different JPEG compression quality. The drawback of this scheme is that it is effective for the case that the tampered region has been compressed with lower quality than its host image. Qu et al. [23] proposed a technique to detect image splicing from image statistics extracted from sharp boundary left by image slicing; the scheme is likely to fail to detect tampered images with the boundaries of tampered regions blurred or undergone some kind of post-processing technique. Dirik and Memon [24] explored artifacts left by Color Filter Array (CFA) for image tampering detection. This scheme is not effective when tampered images have been created from images taken from some specific sensor which does not leave CFA artifacts. Wang et al. [25, 26] derived image statistical features in the spatial domain; in [25], image features are extracted from occurrence matrices of pixels in edge images; in [26], low-dimensional image features

are formulated from the stationary distribution of transition probabilities of pixels in edge images. Sutthiwan et al. [12] presented a natural image model formulated from transition probability matrices of thresholded difference MBDCT arrays and from edge statistics of reconstructed images.

Currently, there are two image datasets for tampering detection evaluation available to the public [27, 28]. The Columbia Image Splicing Detection Evaluation Dataset [27] consists of 933 authentic and 912 spliced grayscale image blocks of size 128×128 pixels. In [14, 18, 19, 21, 29], the efficacies of image features have been evaluated over [27], and the image statistical model in [29] has achieved the relatively highest detection rates (91%). The Institute of Automation at Chinese Academy of Science (CASIA) constructed color image tampering detection evaluation datasets. CASIA Tampered Image Detection Evaluation Database (CASIA TIDE) [28] contains two versions of dataset: CASIA TIDE v1.0, a small-scale dataset, and CASIA TIDE v2.0, a large-scale dataset. The image features proposed in [12, 20, 25, 26] have been evaluated over [28] with rather high detection accuracy rates (all above 90% and even as high as 99%); however, the high detection rates turn out to be not truthful because the dataset [28] inherits some statistical artifacts other than what caused by image tampering, which is to be illustrated later in this chapter.

This chapter introduces an effective scheme for passive-blind image tampering detection under a machine learning framework to distinguish tampered images (positive instances) from authentic images (negative instances). The proposed image statistical features are built upon the analysis of changes of the distribution of block discrete cosine transform (BDCT) coefficients of images. It is conjectured that no matter how profoundly

image tampering has been done, tampered regions are unnatural, exhibiting irregularity causing image statistical properties to deviate from those of natural images. This irregularity may be deemed as noise locally added to the host image (an authentic image). The aforementioned noise may be viewed in the spatial domain, as unnatural changes of pixel values along tampered region(s); in the frequency domain, such a noise may be viewed as alteration in the distribution of BDCT coefficients.

A great diversity of host images, pasted image fragments, image tampering operations yields complicated changes in the correlation pattern of block DCT coefficients. In [20, 29-31], the utilization of block discrete cosine transform with multiple block sizes (MBDCT) has been brought into play under the assumption that a set of BDCT coefficient arrays collectively generated by MBDCT would reflect such changes more effectively than a single-block-size BDCT array. That is, the utilization of multiple block sizes in BDCT would likely enhance the image tampering detectability. Since this idea is analogous to a rake receiver in radio technology which makes use of multiple antennae to mitigate multipath fading to improve the information reliability of received signals, the term “rake transform” (originally from [31]) has been used to represent the MBDCT.

The proposed image statistical features employ rake transform to capture the complicated noise introduced by image tampering with various frequencies. Correlation modeling of BDCT coefficients of the resultant arrays of rake transform has been done by a one-step first-order Markov process. Therefore, the proposed method is referred to as Markovian rake transform. Although rake transform has been used in image feature derivation in [20, 29-31], this chapter written based on [12, 13] presents rake-transform

based features strictly derived exclusively from Markov process for the first time. That is, it utilizes part of the general framework in [29]. The efficacy of image features is evaluated with support vector machines (SVM) over the adjusted CASIA TIDE v.2.0, in which, prior to feature extraction, the images in the dataset have undergone some pre-processing to alleviate non-intrinsic artifacts to ensure a fair performance evaluation.

2.2 Markovian Rake Transform

Markovian rake transform (MRT) applies Markov process (MP) to difference arrays derived from the arrays generated by rake transform. The proposed image statistical features in this chapter differ from those in [20, 29-31] in the following: 1) features are exclusively generated from the application of MP, which is of second-order statistics; 2) three block sizes utilized are 4×4, 8×8, and 16×16. MRT framework can be viewed as a multi-resolution approach based on [32].

2.2.1 Difference MBDCT Arrays (DMBDCT)

The $n \times n$ block discrete transform of an image $f(x,y)$ of size $S_u \times S_v$ results in a 2-D array of the same size as the input image with each consecutive and non-overlapping $n \times n$ block filled up with corresponding block discrete cosine transform (BDCT) coefficients. Rake transform uses a set of different block sizes for BDCT; however, the transform using each block size takes place independently, resulting in multiple BDCT coefficient arrays.

To suit these coefficients to MP modeling, they have to be quantized to integers. For block size 8×8, JPEG coefficients are directly utilized not only because of the wide usage of JPEG images but also because of the superior image tampering detectability of its MP features; BDCT coefficients resulting from other block sizes are rounded to the

nearest integer in this work. A side benefit of the utilization of JPEG coefficients for block size 8×8 is the reduction in computational complexity. Note that an uncompressed image should be JPEG compressed with quality factor equal to 100 before feature derivation.

In MP feature formation [32], the magnitude of quantized BDCT coefficients are taken as input to the process, and in very much the same way, defined are the arrays containing the magnitude of quantized BDCT coefficients as MBDCT arrays.

Effective image statistical features should form distinguishable patterns among different classes with as little influence of image content as possible. Such an influence is, to a certain extent, removed by using difference MBDCT 2-D arrays in feature formation; consequently, transitional characteristics within image are boosted considerably.

Let $F_n(u,v)$, $u \in \{0,1,\ldots,S_u\text{-}1\}$, $v \in \{0,1,\ldots,S_v\text{-}1\}$, denote each of MBDCT arrays generated from a given image with block size $n\times n$, $n \in N$. Note that u and v are discrete variables in the frequency domain of horizontal and vertical axes, respectively.

Intuitively speaking, the transitional characteristics extracted from such difference arrays in horizontal and vertical directions are more informative than those extracted from diagonal difference arrays. In feature formation, only the difference arrays in horizontal and vertical directions are therefore used to keep feature dimensionality not too high. a set of difference arrays of interest derived from MBDCT arrays is denoted as DMBDCT arrays, each of which reads:

$$D_h(u,v;n) = F_n(u,v) - F_n(u+1,v) \tag{2.1}$$

$$D_v(u,v;n) = F_n(u,v) - F_n(u,v+1) \tag{2.2}$$

$u \in \{0,1,\ldots,S_u\text{-}2\}$ and $v \in \{0,1,\ldots,S_v\text{-}2\}$; the difference arrays in horizontal and vertical directions are denoted by $D_h(u,v;n)$ and $D_v(u,v;n)$, respectively.

2.2.2 Thresholding

In [32], it is shown experimentally that the distribution of the elements of the difference arrays derived from JPEG coefficient arrays follows the generalized Laplacian distribution which implies high correlation among these elements. The fact that the distribution these elements are highly dense in the vicinity of zero value brings legitimacy of a thresholding technique which sets the values of such elements falling out of the interval [$-T$, T] to $-T$ or T, correspondingly.

The experiment on part of the dataset [28] unveils that the distributions of the elements of the difference arrays not only of JPEG coefficients but also of integer-rounded coefficients resulting from $n \times n$ BDCT arrays, n = 2, 4, 16, 32, also follow the generalized Laplacian distribution.

The selection of a threshold value T is a subjective issue, critical to feature dimensionality. With a larger T, more energy of the distribution is captured, more powerful resulting features tend to be, and however the dimensionality of features is higher. A reasonable T should capture sufficient image statistics and yield adequately discriminative correlation pattern; in other words, it should provide a suitable balance between feature dimensionality and classification performance.

Table 2.1 Mean (%) Numbers of Elements of Horizontal Difference 2-D Arrays

n	2	4	8	16	32
Mean	59.34	85.90	87.64	96.85	97.02

The statistics in Table 2.1 are computed from authentic images in [28] when T is set to four. For the balance of detection accuracy and computational efficiency, $T = 4$ is chosen to be used in MP feature formation of all the block sizes of interest because for most block sizes it encloses much of the energy of the distributions as well as yields fairly well-performed MP features. Some statistics are shown in Table 2.1.

2.2.3 Transition Probability Matrix

Correlation patterns of DMBDCT arrays can be modeled by a first-order Markov process and characterized by transition probability matrices (TPMs). In this chapter, the elements of DMBDCT arrays in the horizontal and vertical directions are correlation measured by using a one-step TMP [33], provided that the correlation between two elements with more than one element apart is relatively low.

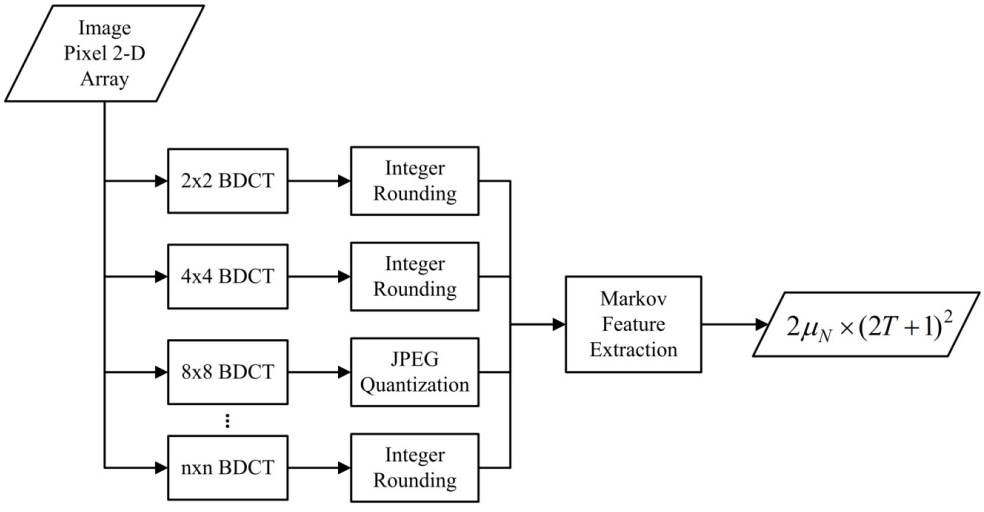

Figure 2.1 Block diagram of generic Markovian rake-transform based feature construction.

Each element of these TPMs associated with the difference arrays in horizontal and vertical directions is defined by (2.3) and (2.4), respectively.

$$p\{D_h(u+1,v;n)=i \,|\, D_h(u,v;n)=j\} = \frac{\sum_{u,v}\delta(D_h(u,v;n)=j, D_h(u+1,v;n)=i)}{\sum_{u,v}\delta(D_h(u,v;n)=j)} \tag{2.3}$$

$$p\{D_v(u,v+1;n)=i \,|\, D_v(u,v;n)=j\} = \frac{\sum_{u,v}\delta(D_v(u,v;n)=j, D_v(u,v+1;n)=i)}{\sum_{u,v}\delta(D_v(u,v;n)=j)} \tag{2.4}$$

where $i, j \in Z$ s.t. $-T \le i, j \le T$; $\delta(x)$ equals one if the statement x is true, or zero otherwise; $\delta(x,y)$ equals one if both of the statements x and y are true, or zero otherwise.

To summarize, the elements of TPMs form MP features, and per block size, the dimensionality of MP features is a function of a threshold value T: $2\times(2T+1)^2$. If μ_N is the cardinality of a set of block size N, the dimensionality of MRT features generated with a fixed T is $2\mu_N\times(2T+1)^2$. A block diagram of generic Markovian rake-transform based feature construction is depicted in Figure 2.1.

2.3 Image Dataset and Necessary Pre-Processing

The official release of CASIA TIDE [28] took place in January 2010. Its official website contains a description of the design principles as well as the structure and image formats. In this section, the relevant description of the dataset is firstly presented. Two issues with the dataset are then pointed out, and finally some necessary pre-processing procedures are described.

2.3.1 Official CASIA TIDE Dataset Description

According to the official web site released in January 2010, CASIA TIDE v1.0 [28] contains 800 authentic images and 925 spliced images of uniform size, 384×256 pixels. These 1,725 images are in JPEG format. All authentic images are divided into eight categories: scene, animal, architecture, character, plant, article, nature and texture.

CASIA TIDE v2.0 [28] contains 7,491 authentic images and 5,123 tampered images of different sizes varying from 240×160 to 900×600 pixels and with different quality factors. Each of these 12,323 images is in one of the following three formats: JPEG, BMP, and TIFF. Indoor category was added into dataset construction. Post-processing such as blurring was applied after crop-and-paste operation by using Adobe Photoshop.

In both CASIA TIDE v1.0 and v2.0 [28] authentic images from all categories were randomly selected to be used in the generation of tampered images which was involved not only cut-and-paste operation but also some geometric transformations, e.g., resizing and rotation. Adobe Photoshop was the tool used to create all the tampered images.

Figure 2.2 Examples of authentic images (all in the top row) and their forgery counter parts (all in the bottom row) [28].

2.3.2 Two Issues

Note that both the experiments in [20] over CASIA TIDE v1.0 [28] and in [12] over CASIA TIDE v1.0 and v2.0 [28] were conducted based on the information available at the initial release of the datasets. In August and September 2010, some technical discussions with the authors of CASIA TIDE v1.0 and v2.0 datasets [28] revealed some information which had not explicitly been included in the official dataset description on the website.

The information is mainly with the following two issues: 1) the JPEG compression applied to authentic images is one-time less than that applied to tampered images; 2) for JPEG images, the size of chrominance components of 7,140 authentic images is only one quarter of that of 2,061 tampered images. Obviously, these uneven processing procedures lead to additional statistical artifacts to that caused by image tampering. This explains very high detection rates reported in the literature recently, including [12, 20, 25, 26].

2.3.3 Keeping Same Number of JPEG Compressions Applied to Tampered and Authentic Images

It is known that Adobe Photoshop defines its own proprietary JPEG compression algorithm. For instance, Impulse Adventure [34] reveals that in JPEG compression, Adobe Photoshop defines its quality range from zero to 12 rather than from zero to 100 as in the standard JPEG compression.

In CASIA TIDE v.2.0 [28], most of the authentic images are from CorelDraw Image Database, the rest are either downloaded or captured by digital cameras, and there are only 54 BMP images among all the authentic images; Adobe Photoshop was used to create 5,123 tampered images, 3,059 of which were saved in TIFF and 2,064 of which were saved in JPEG.

Among all the tampered images saved in JPEG by Adobe Photoshop, about 100 such images were saved with image quality 12, about 10 such images were saved with image quality nine, and more than 1900 such images were saved with image quality eight (which is approximately equivalent to the quality factor 84 [34] in the standard JPEG compression).

Considering that one or more authentic images were used as the input to create tampered images, it is most likely that those tampered images saved in JPEG have undergone one more time JPEG compression than the authentic images in JPEG, given that the chance of a tampered image having been created by using all BMP images is relatively rare. To assuage the side effect on classification performance owing to the difference in the number of JPEG compressions between authentic images and tampered images, Matlab is used for standard JPEG compression to lessen the influence of the difference in the number of JPEG compressions by the following procedure: 1) re-

compress 7,437 JPEG authentic images with quality factor = 84 (approximately equivalent to eight in Photoshop); 2) JPEG compress 3,059 TIFF tampered images by Matlab with quality factor = 84; 3) leave 2,064 JPEG tampered images untouched. After the completion of the above-described measures, randomly selected 5,000 pairs of authentic images and tampered images are used for the experiments to be discussed in Section 2.4.

2.3.4 Keeping Same Treatment to Chrominance Channels

When an image saved in JPEG at image quality seven and higher by Adobe Photoshop, there is no sub-sampling in image chrominance components, hence, encoding a larger array of color component than what standard JPEG compression does. That is, image chrominance components of most of these tampered images have not undergone sub-sampling. As a result, most tampered JPEG images in CASIA TIDE [28], color channel selection for feature extraction must be handled with care to circumvent any preferential treatment of color component.

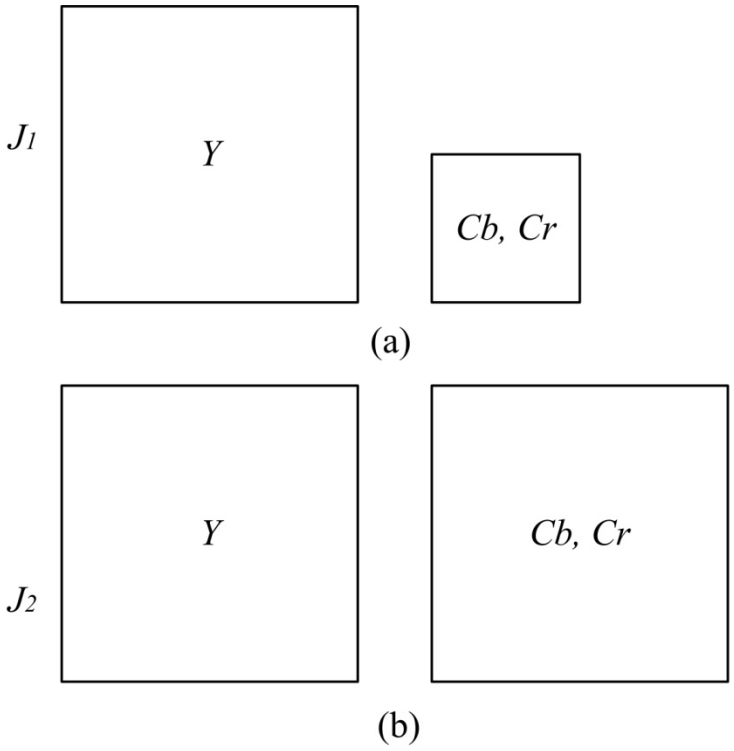

Figure 2.3 Relative sizes of JPEG coefficient arrays: (a) with chrominance sub-sampling (4:2:0); (b) without chrominance sub-sampling (4:4:4).

As can be seen in Figure 2.3, the size of JPEG coefficient arrays of chroma channels (*Cb* and *Cr*) with sub-sampling is only one-fourth of that of no sub-sampling.

In this subsection, a simple experiment is presented to show the difference in image statistical properties that the chrominance component with and without sub-sampling can bring out. The three-step outline of the experiment is as follows.

First, ucid00017.tif, randomly selected from UCID dataset [35], is compressed into J_1 by standard compression (4:2:0) with quality factor equal to 95. J_1 is further compressed into J_2 by Adobe Photoshop with image quality equal to 11. According to [34], in Adobe Photoshop, JPEG compression with image quality = 11 is virtually equivalent to standard JPEG compression with quality factor = 95; their major difference

lies in chrominance sub-sampling: J_1 has been chrominance sub-sampled by a factor of two while J_2 has not.

Second, the edge images of each component in YCbCr color space of J_1 and J_2 are generated according to the process described in Figure 2.4 in which the edge detector calculates the horizontal difference array of an input image.

Third, for each edge image, first four order statistics of pixel intensities are computed and tabulated in Table 2.2. Note that μ stands for mean, σ^2 for variance, γ_1 for skewness, γ_2 for kurtosis.

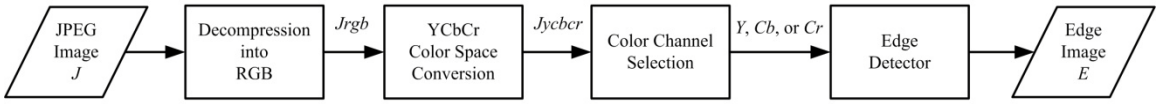

Figure 2.4 Block diagram of edge image generation.

Table 2.2 Image Statistics of the Edge Images Generated from ucid00017.tif

Channel	Image	μ	σ^2	γ_1	γ_2
Y	J_1	11.96	197.09	2.39	11.04
	J_2	11.90	195.22	2.34	11.11
Cb	J_1	0.87	1.35	2.85	18.55
	J_2	1.20	3.60	4.42	37.62
Cr	J_1	0.79	1.03	2.16	11.70
	J_2	1.17	2.89	3.20	19.80

In Table 2.2, J_1 is the JPEG compressed version by using Matlab, and J_2 is obtained by JPEG compressing J_1 with an almost equivalent quantization step using Adobe Photoshop image without chrominance sub-sampling. For details refer to the outline of experiment presented prior to Figure 2.4. Figure 2.5 displays J_1 and J_2 generated from the test image and their corresponding edge images. Note that all the edge images are logarithm transformed to enhance edge visibility.

The following conclusions can be drawn from Table 2.2 and Figure 2.5 on the obtained statistical properties of edge images: 1) for the first and second order statistics derived from image luminance (*Y*), they are, in any case, all highest among three channels: and all four order statistics of both JPEG images are close; 2) the first four order statistics of edge images from image chrominance (*Cb* or *Cr*) derived from J_2 are significantly different from those from J_1; 3) the major objects in the image are more discernible in the edge images derived from *Cb* or *Cr* than those from *Y*.

In this experiment, a given image has not yet gone through any tampering. Instead, it has gone through common image processing operations, i.e., compression and color channel processing. On the one hand, chrominance component sub-sampling has been done to the image as shown in (a), (c), (e), (g); on the other hand, no chrominance component sub-sampling has been so as shown in (b), (d), (f), (h). It is shown that the first four order statistics of the edge information from chroma channels of these two images are rather different. This experiment could partly explain the high detection rates reported [25, 26] by image features extracted from image chrominance, which is much higher than that from *Y* component. For image feature extraction in CASIA TIDE v.2.0 [28], *Y* channel in which no bias in color information presents between Matlab and Photoshop software has been chosen as the only input.

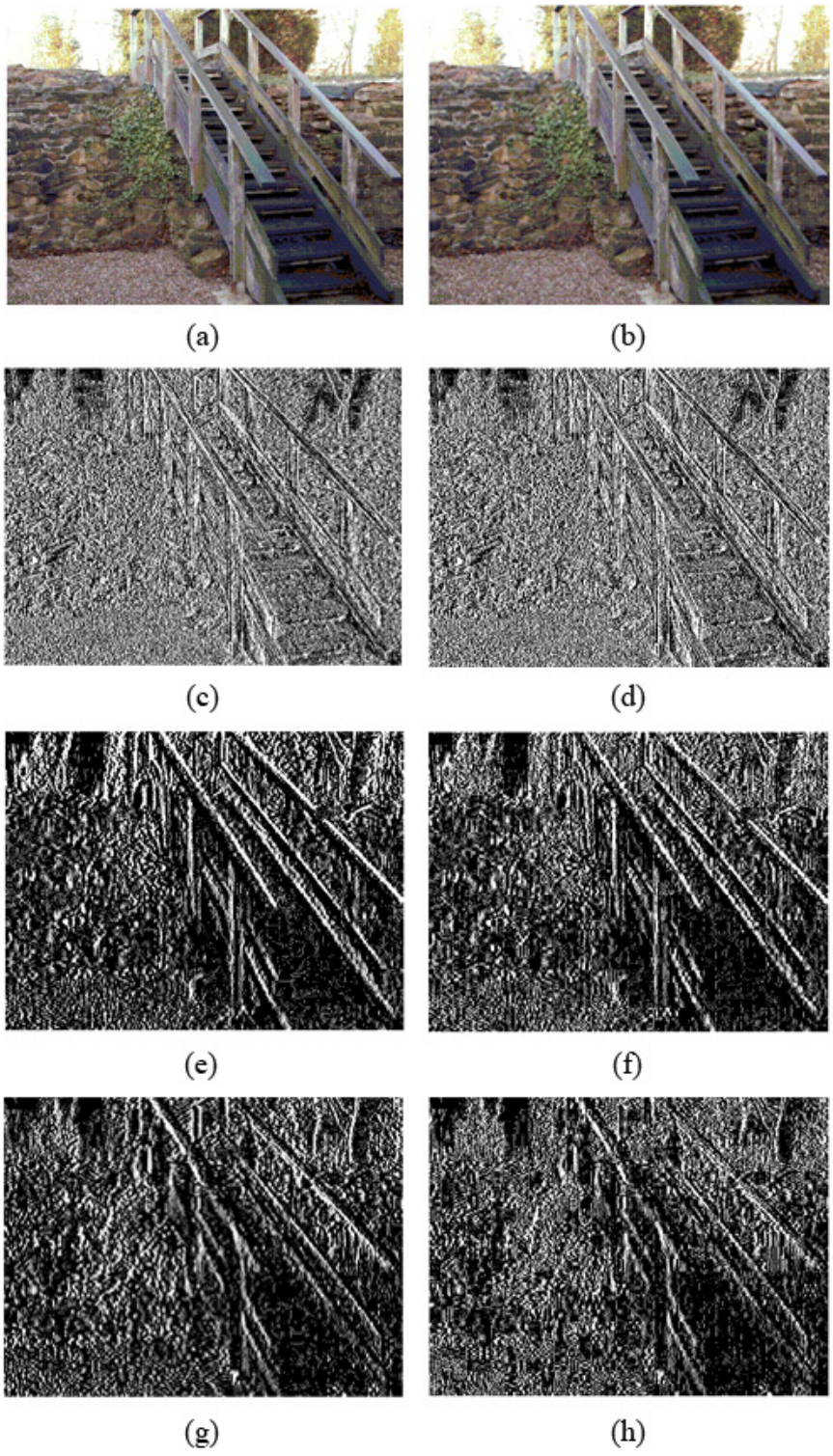

Figure 2.5 Comparison among edge images. (a) Image J_1; (c), (e), and (g) are the corresponding edge images of *Y*, *Cb*, and *Cr* of (a), respectively; (b) Image J_2; (d), (f), and (h) are edge images of (b) from *Y*, *Cb*, and *Cr*, respectively.

2.4 Experimentation

In tampering detection, tampered and authentic images are considered positive and negative instances, respectively. Throughout all the experiments in this chapter, the support vector machines (SVM) are employed, the Matlab codes of which are available at [36], with degree two polynomial kernel. The results reported in Tables 2.3 and 2.4 are the average detection rates of 20 independent iterations of SVM classification with random data partition (five-sixths for training and one-sixth for testing). There is no performance comparison the performance by using the proposed features with that reported in [25, 26] because the reported classification performances there are inaccurate, owing to the two facts analyzed in Section 2.3. As discussed in Section 2.3, some measures have been taken to suppress the unsolicited artifacts in CASIA TIDE v.2.0. before using it to evaluate the proposed features as well as the features in [29] for performance comparison.

2.4.1 Empirical Model Validation and Classification Performance

In model validation, hierarchical search is employed to determine a proper choice of block sizes to be used in the generation of Markovian rake-transform (MRT) features through some possible combinations of set $N = \{2, 4, 8, 16, 32\}$. For all block sizes, the threshold value $T = 4$ is chosen, resulting in the dimensionality of features associated with an individual block size equal to 162.

It is obvious from the information contained in Table 2.3 that the MP features generated with block sizes 2×2 and 32×32 perform poorly. This fact eventually led to the exclusion of these two block sizes out of the choice. Table 2.4 shows the performances of MRT features generated with all the possible combinations of the leftover block sizes.

To conclude, a set of block sizes {4×4, 8×8, 16×16} is recommended for MRT feature formation for tampering detection over CASIA TIDE v2.0 [28]. This is not only because the information shown in Table 2.4 suggests itself but also because some extended experiments reveals that adding more block sizes into the suggested combination barely improves the performance of the resultant MRT features, while increases feature dimensionality and the computational complexity. The proposed 486-D features are abbreviated in this chapter as MRT-486 while 266-D features in [29] as MBDCT-266.

Table 2.3 Detection Rates of Individual MP Features

n	2	4	8	16	32
TP Rate	66.99%	69.10%	63.05%	64.03%	65.78%
TN Rate	69.05%	73.16%	91.49%	84.20%	64.59%
AC	68.02%	71.13%	77.27%	74.12%	65.18%
Dimensionality	162	162	162	162	162

Table 2.4 Detection Rates of MRT Features

n	4,8	4,16	8,16	4,8,16
TP Rate	68.19%	71.48%	66.75%	72.43%
TN Rate	89.41%	82.27%	90.28%	87.05%
AC	78.80%	76.87%	78.51%	79.74%
Dimensionality	324	324	324	486

Depicted in Figure 2.6 are the receiver operating characteristic (ROC) graphs along with their corresponding areas under curve (AUCs) of the three classification systems built from individual MP features generated by using one of the three block sizes constituting the suggested combination as well as that built from MRT-486. Not only ROC graphs but also AUCs indicate the improvement in the ability of classifier trained by the proposed MRT features over those trained by individual MP features.

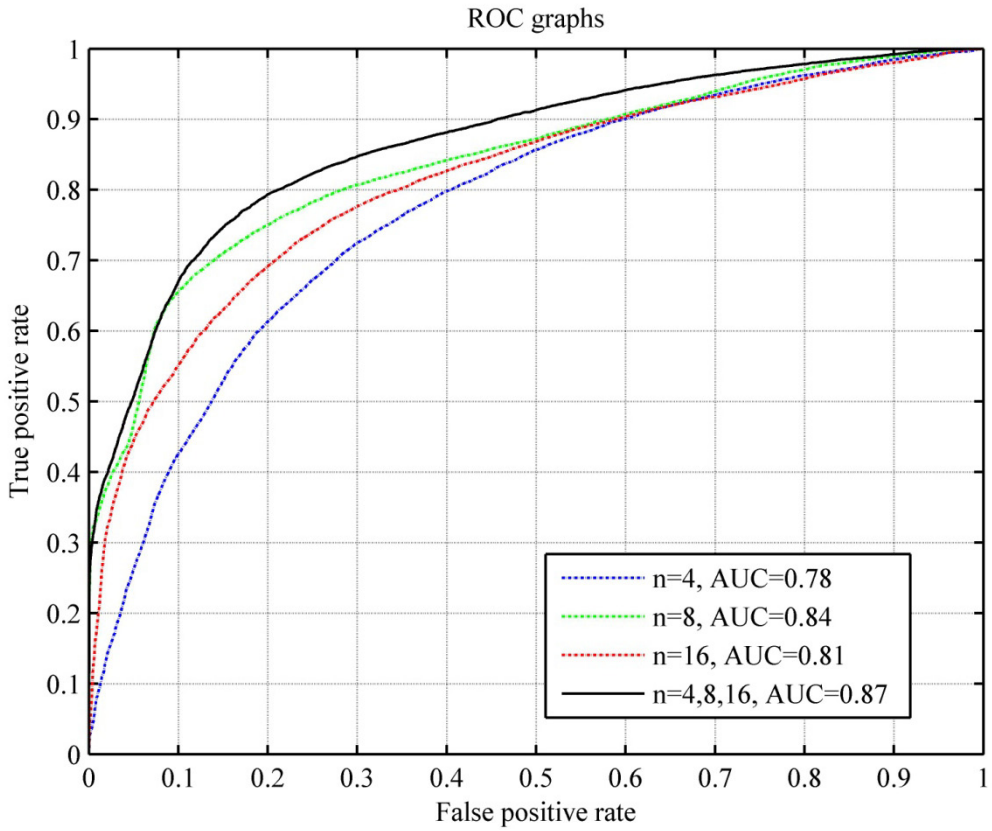

Figure 2.6 ROC graphs and AUCs of MRT-486 (solid line) and of the three associated individual MP features.

MBDCT-266 features are also evaluated over the adjusted dataset under the same classification setting, and the performance comparison is presented in Table 2.5 and reveals that the proposed MRT-486 features perform slightly better than MBDCT-266 features. Although the classification performances between MRT-486 and MBDCT-266 features do not differ significantly, the advantage of MRT-486 over MBDCT-266 features is its much lower computational complexity which is reflected in the required time to compute a feature vector. In HP Pavilion dv6930us, less than two seconds are required to compute an MRT-486 feature vector whereas more than 26 seconds are needed in the generation of an MBDCT-266 feature vector.

Table 2.5 Performance Comparison of MRT-486 and MBDCT-266

Feature Type	MRT-486	MBDCT-266
TP Rate	72.43%	75.83%
TN Rate	87.05%	83.12%
AC	79.74%	79.48%
Dimensionality	486	266

2.4.2 Illustration of Feature Separability between Tampered and Authentic Images

Figure 2.7 depicts feature separability of MRT-486 on 2-D projection. Principle Component Analysis (PCA) is employed in the feature vector projection.

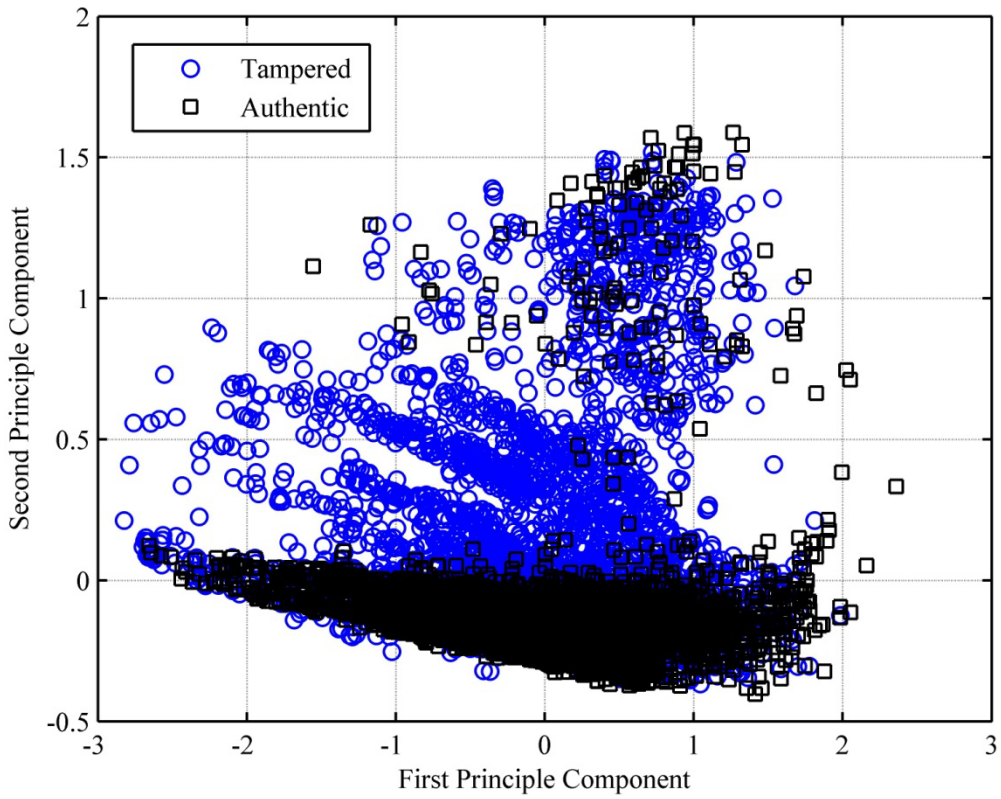

Figure 2.7 PCA 2-D projection of MRT-486.

2.5 Tests on Real Images Outside the Dataset

Practical value of any classification system is desirable, and tests on instances outside the dataset used could be one of the ways to evaluate such value. The 20 classifiers resulting from the 20 independent iterations of SVM classification are used to test some authentic images and tampered images outside the dataset. These images were downloaded from the Photo Tampering through History [37], exemplifying image tampering in real life. The majority voting from the 20 classifiers is used to derive a decision on these images. Basically the ground truth of such images is available, i.e., it is known whether a test image is authentic or tampered. However, in most cases, no explicit knowledge on how and where image tampering has been done can be obtained.

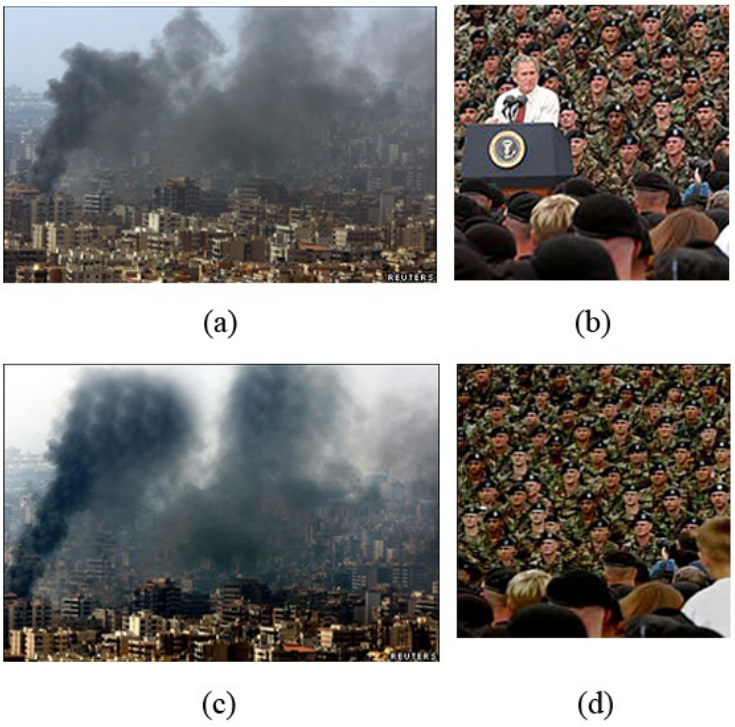

Figure 2.8 Examples of real-life tampered images. In the top row: (a)-(b): (a) the authentic image of Israel air raid on Lebanon; (b) the authentic image of George W. Bush. In the bottom row (c)-(d): (c) the forgery counterpart of (a) ; (d) the forgery counterpart of (b). (adapted from [37]).

Table 2.6 Test Results on Figure 2.8

Figure	Type	P	N	Final Decision
2.8a	Authentic	0	20	Authentic
2.8b	Authentic	20	0	Tampered
2.8c	Tampered	19	1	Tampered
2.8d	Tampered	20	0	Tampered

In Table 2.6, P (tampered) and N (authentic) are the numbers of trained classifiers that decide a given test image as authentic image and tampered, respectively. Among more than twenty collected images from outside the dataset, the accuracy is far falling behind that of the tests over images inside the adjusted CASIA TIDE v.2.0 [28]. Figure 2.8 and Table 2.6 detail the testing results of four test images outside the dataset.

2.6 Discussion and Conclusions

In this chapter, an image statistical framework in the frequency domain based on Markovian rake transform (MRT) is introduced on image luminance. The MRT applies Markovian process to difference 2-D arrays independently derived from block discrete cosine transform with multiple block sizes. MRT framework is a sub-framework within the general natural image framework proposed in [29].

The efficacy of image features is assessed over 5000 image pairs from CASIA TIDE v.2.0 [28], a large-scale dataset with some particular measures taken to eliminate unsolicited artifacts left during the JPEG compression. The analysis and procedure adopted are presented in Section 2.3.

The proposed MRT features of size 486, abbreviated as MRT-486, consists of Markov process based features generated with three different block sizes: 4×4, 8×8 (utilizing JPEG coefficients), and 16×16. Image luminance is chosen for feature

extraction because it does not suffer from any uneven treatment applied to color information.

Although the dimensionality of MRT-486 seems high, they actually require relatively low computational time because only rather simple operations are involved in feature extraction. The average time of feature extraction on HP Pavilion dv6930us with un-optimized Matlab codes for one image over the dataset is less than 2 seconds, while that of the features in [29] is more than 26 seconds.

Even with low complexity, MRT-486 perform fairly well, the accuracy of almost 80% over 5,000 pairs of authentic and tampered images randomly drawn from the adjusted CASIA TIDE v.2.0 after careful adjustments. The area under ROC curve (AUC) is about 0.87. Moreover, the proposed features outperform any combination of MP features, provided that the number of block sizes utilized in MRT is less than or equal to three. Such distinction in the performance of the proposed MRT features is shown graphically in the ROC graphs in Figure 2.6.

The tests on images outside the dataset used, although the success rate is still far from satisfactory, unveil the practicality of the proposed natural image model to a certain extent. This indicates that image tampering detection is still encountering enormous challenges, especially of the tests on real-life tampered images.

CHAPTER 3

CLASSIFICATION OF COMPUTER GRAPHICS AND PHOTOGRAPHIC IMAGES

In this chapter, three frequency-domain image features are introduced to identify computer graphics by employing second-order statistics of the frequency domain to capture the significant statistical difference between computer graphics and photographic images. The first kind of features relies on Markov statistic modeling on JPEG information from multiple color channels in YCbCr color space. The second kind of features utilizes Boosting Feature Selection (BFS) to reduce the dimensionality of the first kind of features without sacrificing the machine learning based classification performance. The third kind of features utilizes Markovian Rake Transform (MRT) features introduced in Chapter 2. The empirical validation reveals that features derived from a combination of two block sizes and from *Y* channel perform better than those derived by the first and second techniques. This experimental result confirms the universality as well as the validity of MRT framework. The related works were published in [38, 39].

3.1 Introduction

Computer graphics (CG) have become more and more photorealistic due to the advancement made in rendering software. As a result, it has become very much difficult for people to visually differentiate them from photographic images (PG).

Therefore, high photorealistic CG may be exploited as either a convincing form of image forgery or a replacement of hard-oriented scene in movie production;

consequently, identifying CG appears to be an important task in both image forgery detection and a benchmark for rendering software. From a practical point of view, an automatic classification system is certainly more suitable and realizable to deal with this issue than human inspection which is hard to reach high accuracy, confidence and reliability in identifying high photorealistic CG.

The objective of this research is to develop a statistical model-based approach to automatically discriminate CG from PG. The effectiveness of the image feature vector is then evaluated by the classifier in the machine learning (ML) framework.

In the literatures, several classification systems based on different types of image features have been reported. Ianeva et al. [40] modeled characteristics of cartoon plays a crucial role in classifying cartoons, a particular type of CG. The features are extracted from the color saturation, color histogram, edge histogram, compression ratio, pattern spectrum and the ratio of image pixels with brightness greater than a threshold, resulting in the image feature vector of size 108. S. Lyu and H. Farid [41] utilized image decomposition based on separable quadrature mirror filters (QMFs) to capture regularities inherent to photographic images. Half of features, of size 36, are derived from the first four order statistics of horizontal, vertical and diagonal high-frequency sub-bands. The other half of features is collected from the same four statistics of the linear prediction error for the high-frequency coefficients. The total number of features is 72 per RGB color channel, yielding a grand total of 216 features. Ng et al. [42] proposed to use 192 geometry features extracted by analyzing the differences existing between the physical generative process of computer graphics and photographic images and characterized by differential geometry and local patch statistics. Chen et al. [43] used

moments of characteristic functions of wavelet sub-bands are used with HSV and YCbCr [44] color models. Each color component image is decomposed into three levels using Haar wavelet. At each level i, $i = 1, 2, 3$, there are four sub-bands (LL_i, HL_i, LH_i, HH_i). Totally, there are 13 sub-bands involved in the feature extraction if the component image itself is deemed to be a sub-band at level zero. For each sub-band, the first three moments are computed, resulting in 39 features. In addition to the component image, its prediction error image is applied to the same process to reduce the influence of image content, so a total of 78 features are the output of a color component. Both two and three color component images are used and reported in [43, 44] which correspond to 156 and 234 features, respectively.

3.2 JPEG-based Markovian Features

There are fundamental boundaries between CG and PG: first of all, fewer colors are contained in CG; secondly, in texture area CG are smoother in color than PG; in CG, the changes in color intensities from one region to another are more abrupt than those in PG; last but not least, edge and shade of CG hold different characteristics from PG. These different image properties are well reflected in the frequency domain. Under these circumstances plus the fact that most of images are in JPEG format, the utilization of the TPM derived from applying MP to model difference JPEG 2-D arrays are proposed to formulate the distinguishing features, which is a second order statistic by its nature. The statistical image features discussed in this section is a variation of the steganalyzer in [32].

3.2.1 JPEG 2-D Array

Owing to wide availability of JPEG images, JPEG 2-D array [32] can be considered as an appropriate input of feature formation. For a given image, this 2-D array has the same size as the given image with each consecutive and non-overlapping 8×8 block filled up with the corresponding quantized block DCT (BDCT) coefficients. The absolute value of each coefficient is measured and used to form the 2-D array shown in Figure 3.1 which is called JPEG 2-D array in this sequel.

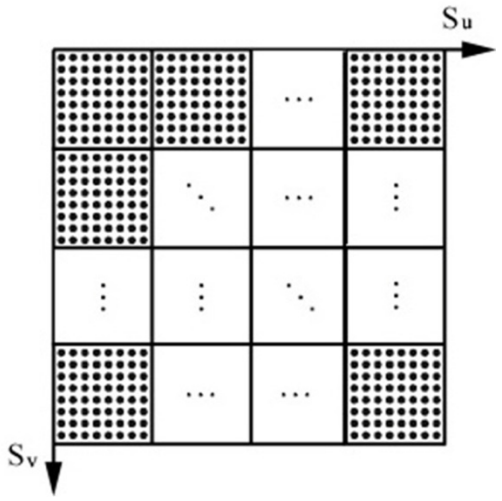

Figure 3.1 Sketch of JPEG 2-D array [32].

3.2.2 Difference JPEG 2-D Array

The influence by the content of various images on CG classification can be reduced largely by using difference JPEG 2-D array [32] which reveals transitional characteristics of image contents.

Denote the JPEG 2-D array generated from a given image by $F(u,v)$, $u = 0, \ldots, S_u\text{-}1$, $v = 0, \ldots, S_v\text{-}1$, where S_u is the size of the 2-D array in horizontal direction and S_v is the size of such an array in vertical direction. As mentioned before, only the difference

arrays along horizontal and vertical directions are utilized here and such difference arrays are formulated as mathematically described in Equations 3.1 and 3.2.

$$F_h(u,v) = F(u,v) - F(u+1,v) \tag{3.1}$$

$$F_v(u,v) = F(u,v) - F(u,v+1) \tag{3.2}$$

where $u = 0, \ldots, S_u\text{-}2$, $v = 0, \ldots, S_v\text{-}2$, and $F_h(u,v)$ and $F_v(u,v)$ denote the difference 2-D arrays along the horizontal and vertical directions, respectively.

3.2.3 Thresholding Difference JPEG 2-D Array

A statistical analysis reported in [32], for a different scenario though, has shown that the distribution of the elements of the difference JPEG 2-D arrays is Laplacian-like, concentrated on zero. This fact leads to the legitimacy of applying a thresholding technique, which sets the difference values falling out of the interval $[-T,T]$ to $-T$ or T, correspondingly.

Due to the fact that the experimental work in [32] was conducted on more than 7,000 images, the selection of T there, would statistically be meaningful in the image set used in this research work. Hence, $T = 4$ is chosen which only about 8% of the elements of difference JPEG 2-D arrays are out of interval [-4,4]. Therefore, with this selected threshold, reasonably sufficient statistical properties of images are captured with manageable computational complexity, which will become clear in Section 3.2.4.

3.2.4 Transition Probability Matrix

It is well known that MP is a statistical tool to model the correlation between elements on an array, and the associated TPM can characterize MP. In this work, difference JPEG 2-D arrays in vertical and horizontal directions are modeled by MP. There are so-called one-step TPM and n-step TPM [33]. Due to the fact that the correlation between two elements separated by more than one element is low, to maintain manageable computational complexity, the one-step TPM is employed in this work, and its graphical representation is shown in Figure 3.2.

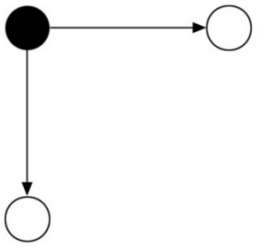

Figure 3.2 Formation of transition probability matrices (adapted from [32]).

The elements of these two matrices associated with the horizontal and vertical difference 2-D arrays are given by

$$p\{F_h(u+1,v)=n \,|\, F_h(u,v)=m\} = \frac{\sum_{u,v}\delta(F_h(u,v)=m, F_h(u+1,v)=n)}{\sum_{u,v}\delta(F_h(u,v)=m)} \tag{3.3}$$

$$p\{F_v(u,v+1)=n \,|\, F_v(u,v)=m\} = \frac{\sum_{u,v}\delta(F_h(u,v)=m, F_h(u,v+1)=n)}{\sum_{u,v}\delta(F_h(u,v)=m)} \tag{3.4}$$

where $m,n \in \{-T,\ldots,0,\ldots,T\}$, the summation range for u and v is respectively from zero to S_u-2 and from zero to S_v-2, and

$$\delta(A=m,B=n)=\begin{cases}1 & \text{if } A=m \,\&\, B=n\\ 0 & \text{otherwise}\end{cases} \tag{3.5}$$

In sum, the feature vector of dimensionality $(2T + 1)\times (2T + 1)\times 2$ is formulated per input channel. With $T = 4$, 162 features are generated per color component. The feature construction procedure is summarized in Figure 3.3.

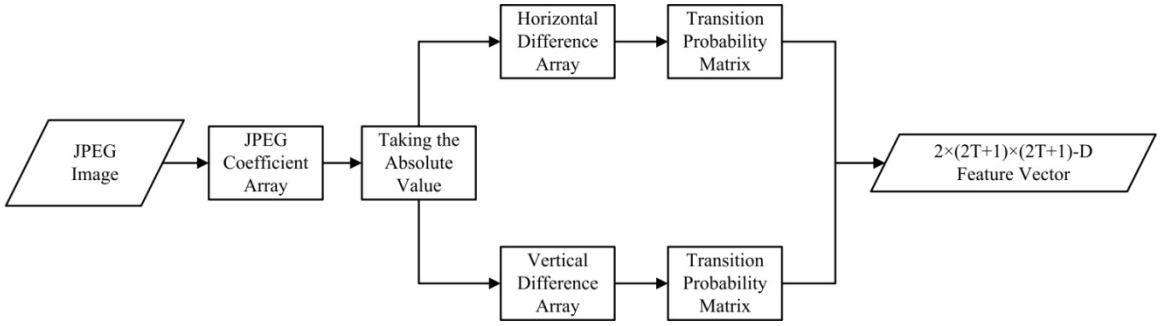

Figure 3.3 Block diagram of feature formation.

3.2.5 Correlation Analysis Among Color Components

Simple analysis is taken and reveals an interesting fact about the correlation among the features generated from different color components. The correlation level is measured by the correlation coefficient $\rho_{x,y}$ [33] between two random vectors x and y whose expected values are μ_x and μ_y and standard deviations are σ_x and σ_y, respectively.

$$\rho_{x,y}=\frac{\operatorname{cov}(x,y)}{\sigma_x\sigma_y}=\frac{E((x-\mu_x)(y-\mu_y))}{\sigma_x\sigma_y} \tag{3.6}$$

Table 3.1 Average Correlation Coefficient Values of Feature Vectors

Channel Combination	CG	PG
Y,*Cb*	0.60	0.66
Y,*Cr*	0.61	0.67
Cb,*Cr*	0.95	0.95

Note that the order of feature vectors in ρ does not affect its value. Statistics shown in Table 3.1 are computed from randomly selected 1,000 CG and 1,000 PG. It implies that features derived from *Cb* and *Cr* components are much more strongly correlated than any other combinations of two components: 1) *Y* and *Cb*; 2) *Y* and *Cr*. This fact, along with the intensive experiments confirms that using all of features constructed from three color components does not improve the feature effectiveness significantly but rather increases computational complexity drastically. In the image feature extraction process, *Y* and *Cb* channels are therefore selected, having the least correlation in pair-wise feature vectors and resulting in 324-D feature vector. However, it is empirically observed that the degree of correlation between features from *Y* and *Cb* are generally close to that between features from *Y* and *Cr*; therefore, it is also legitimate to choose *Y* and *Cr* combination for feature extraction.

3.3 Boosting Feature Selection

It is well known that discrete AdaBoost [45], a learning algorithm, enhances binary classification performance. The logic behind this algorithm is to combine a set of weak classifiers (weak learners) to form a strong classifier in a weighted manner.

Given a set of training data $(x_1, y_1), \ldots, (x_m, y_m)$ with a variable of the feature vector x_m and its corresponding label $y_m \in \{-1, 1\}$, where $m = 1, \ldots, M$. (+1 denotes the positive samples and -1 the negative ones), one can define:

$$F(x) = \sum_{i=1}^{M} c_m f_m(x) \tag{3.7}$$

The outline of discrete AdaBoost algorithm is as follows:

1. Start with weights $\omega_i = 1/N$, $i = 1, \ldots, N$.
2. Repeat for $m = 1, \ldots, M$.

   2.1 Fit the classifier $f_m(x) \in \{-1,1\}$ using weights ω_i on the training data.

   2.2 Compute $err_m = E_\omega[1(y \neq f_m(x))]$ and then $c_m = \log[(1- err_m)/err_m]$

   2.3 Set $\omega_i \leftarrow \omega_i \cdot \exp[c_m \cdot 1(y_i \neq f_m(x_i))]$, $i = 1, \ldots, N$, and re-normalize it so that $\sum_i \omega_i = 1$.
3. Output the classifier $sign[\sum_{m=1}^{M} c_m f_m(x)]$.

Tieu and Viola [46] introduced BFS algorithm combining AdaBoost and ensemble feature selection together. The effective features for classification are distinctively selected on the basis of lowest weighted error err_m for the given weighted ω_i. As the weight changes, different input variables are selected for the current classifier $f_m(x)$. Mathematically, one can denote:

$$f_m(x) = \beta_m b(x, \gamma_m) \tag{3.8}$$

where β_m is a multiplier and γ_m is the order of dimensionality of x in the whole input vector, and $b(.)$ denotes the m^{th} column of the input feature vector. In [46], it is stated that one can solve for an optimal set of parameters through a "greedy" forward stepwise approach with updates which can be mathematically defined as in Equation 3.9.

$$\{\beta_m, \gamma_m\} \leftarrow \arg\min_{\beta,\gamma} E[y - F_{m-1}(x) - \beta b(x; y)]^2 \tag{3.9}$$

for $m = 1, 2, \ldots, M$ in cycles until convergence, where $\{\beta_m, \gamma_m\}$, $m = 1, \ldots, M\text{-}1$, are fixed at their corresponding solution values at earlier iterations in the algorithm. After several iterations, a powerful classifier could be produced using only a portion of all input variable.

3.4 Image Dataset

Computer graphics (CG) are collected from [47, 48] where more than 50 pieces of rendering software, e.g., 3D Studio Max, After Effects and AutoCad, were used to generate CG. For photographic images (PG), a small portion of the image sets are also from [49] while the majority is gathered by the previous members of Prof. Yun Shi's research group. There are totally 3,000 CG and 3,000 PG. The Q-factors of PG range from 75 to 100 and those of CG from 65 to 98.

Figure 3.4 Examples of computer graphics in the dataset.

Figure 3.5 Examples of photographic images in the dataset.

3.5 Experimentation

In this section, three kinds of image features generated are evaluated over the same dataset which allows a direct comparison among the features of interest.

The first kind of features is 324-D Markovian features, denoted as MP-324, derived from JPEG information (single BDCT) and two color channels as described in Sections 3.2.

The second kind of features is formed by applying Boosting Feature Selection (BFS) to MP-324 to select 150 features as the number of selected features is empirically

found to be optimal within the searching range. The second kind of features is then denoted as MPBFS-150.

The third kind of features is derived based on Markovian Rake Transform (MRT) framework discussed in Chapter 2. By brute force searching, the best combination of block sizes {4×4, 16×16} can be determined which also results in the feature dimensionality of 324. It is also experimentally found that increasing the number of block sizes used to form feature barely improves classification performance. This type of features, denoted as MRT-324, is extracted only from *Y* channel in order to control the feature dimensionality not to be too high.

Support vector machines (SVM) [36] with RBF kernel is used to evaluate these three types of features. The kernel parameters are determined by coarse-grid searching with six-fold cross validation. The SVM classifiers are trained independently for 20 runs. Each run, 2500 pairs are randomly selected for training and the leftover 500 pairs for testing. The average detection rates are shown in Table 3.2. Note that in this chapter computer graphics and photographic images are considered positive and negative instances, respectively.

Table 3.2 Performance Comparison of MP-324, MPBFS-150, and MRT-324

Feature Type	MP-324	MPBFS-150	MRT-324
TP Rate	93.65%	93.72%	96.61%
TN Rate	94.72%	94.68%	95.88%
AC	94.19%	94.20%	96.25%
Dimensionality	324	150	324

3.6 Discussion and Conclusions

In this chapter, three different kinds of features based on the application of Markov process in the frequency domain are introduced to identify computer graphics from photographic images.

The first kind of features (MP-324) are derived by applying Markov process (MP) to model difference JPEG 2-D arrays along horizontal and vertical directions. The distinguishing features are derived from all elements in one-step transition probability matrices (TPM) which characterize the MP. Applying the MP additionally to difference JPEG 2-D arrays along major and minor diagonal directions and deriving features from two-step or multiple-step TPM would increase classification performance at the expense of computational complexity. The correlation analysis in Section 3.2.5 suggests that MP features from two chroma channels are highly correlated which is undesirable, so it is legitimate to select only one chroma channel (*Cb* or *Cr*) together with image luminance (*Y*) for feature extraction. In this work, the combination of *Y* and *Cb* is selected due to the lowest average correlation coefficients of the extracted features.

The second kind of features is derived by applying Boosting Feature Selection (BFS) technique to select some effective features. The resultant classification performance remains the same, even though feature dimensionality has been greatly reduced. It is extensively empirically found that BFS can hardly boost the classification performance of many frequency-domain Markovian-based features. Consequently, if the number of training patterns is sufficient, it is not recommended to employ BFS or any other feature reduction techniques that do not enhance the classification performance as such techniques themselves increase the computational complexity.

The third kind of features is derived based on Markovian Rake Transform (MRT) framework introduced in Chapter 2. MRT features of size 324, abbreviated as MRT-324, are derived from using two different block sizes: 4×4 and 16×16. MRT-324 is remarkably more effective than MP-324 and MPBFS-150. Both MP-324 and MPBFS-150 are actually derived based only on a single block-size DCT while MRT-324 is on multiple-block-size DCT. In a nutshell, the experimental work in this chapter confirms not only the universality but also the effectiveness of MRT framework over MP framework.

CHAPTER 4

ANTI-FORENSICS OF DOUBLE JPEG COMPRESSION DETECTION

In Chapter 4, a simple yet effective anti-forensic scheme capable of misleading double JPEG compression detection techniques is presented according to [50]. Based on image resizing with bilinear interpolation, the proposed operation aims at destroying JPEG grid structure while preserving reasonably good image quality. Given a doubly compressed image, the proposed attack modifies the image by JPEG decompressing, shrinking and zooming the image with bilinear interpolation before JPEG compression with the same quality factor as used in the given image. The efficacy of the proposed scheme has been evaluated on two prominent double JPEG detection techniques and the outcome reveals that the proposed scheme is mostly effective, especially in the cases that the first quality factor is lower than the second quality factor. The related work was published in [50].

4.1 Introduction

The advances of digital cameras as well as image editing tools have made digital images new form of statue of memory. Duplication, distribution, or tampering of such media can be easily done which calls for the necessity to be able to trace back the authenticity or history of media. Digital image forensics is a branch of research that aims to resolve the imposed problem. As sophisticated as digital image forensic schemes have been designed on the one hand, there are weak points of such schemes on the other hand. Over the past few years, anti-forensics has emerged as a relatively new branch of research. It aims at

revealing the weakness of forensic technology and this may lead to improve the next generation of such a technology.

Double JPEG compression detection has been of great significance to digital image forensics, especially to image steganalysis and tampering detection. Its goal is to distinguish between JPEG images compressed once and those compressed twice. In this chapter, an anti-forensic technique is introduced and its effectiveness is evaluated on two highly effective double JPEG compression schemes.

Stamm et al. [51] proposed an anti-forensic scheme to hide evidence of JPEG compression by adding adjustable noise to DCT coefficient of JPEG compressed images and removing blocking artifacts was proposed. It has been shown that the anti-forensically modified images have statistical properties close to those of uncompressed images. Stamm et al. [52] proposed de-blocking operation on top of the scheme introduced in [51] to fool forensic schemes that rely on the evidence of JPEG compression. Stamm et al. [53] also indicated that the proposed scheme could potentially mislead double JPEG compression schemes and extensively presented the afore-mentioned anti-forensic schemes.

4.2 Double JPEG Compression Detection

Double compression artifacts, also known as double quantization (DQ) artifacts, can be effectively traced by analyzing the statistical properties of DCT coefficients and are characterized by peak-and-valley pattern in a JPEG mode histogram. According to Popescu [54], double compression introduces periodic peak artifacts to a JPEG mode histogram when the ratio of second quantization step (q_2) over the first one (q_1) is not an

integer; otherwise, the mode histogram of singly compressed and doubly compressed images would be indistinguishable. In JPEG compression, the expected quality of compressed images is user-defined and defined by a quality factor (QF) which is an 8×8 matrix filled up with 64 quantization steps; therefore, in the generation of a doubly compressed image, a pair of first quality factor (QF1) and second quality factor (QF2). The aforementioned condition of periodic artifacts implies that the detection of double JPEG compression of doubly compressed images generated by QF1 = QF2 is intrinsically undetectable with this approach. Note that some new strategy has made this detection doable [55].

Several double JPEG compression detection schemes have been adopted under the passive-blind framework; however, in this chapter, two rather efficient schemes are picked up to be attacked: 1) Chen et al. [56] proposed 324-D feature vector derived from applying Markov process to horizontal, vertical, major diagonal and minor diagonal difference JPEG 2-D arrays. Herein, the 324-D feature in [56] is abbreviated as MP-324. Owing to the limitation in the size of image dataset to be used, MP-324 is modified by ignoring the features generated from both diagonal difference arrays which results in the feature dimensionality reduction from 324 to 162. The modified feature scheme is therefore abbreviated as MP-162. The efficacy of image features have been evaluated by Support Vector Machine [36]; 2) Bin et al. [57] utilized the probabilities of the first digit of 20 quantized JPEG AC modes to form a 180-D feature vector which is abbreviated in this chapter as MBFDF. The effectiveness of image features is assessed by Fisher's Linear Discriminant (FLD).

Per pair of the first quality factor (QF1) and the second quality factor (QF2), both double JPEG compression detection schemes are re-evaluated using the corresponding classification settings over 1,338 pairs of singly and doubly compressed images in gray-scale of size 384×512 (landscape) or 512×384 (portrait). These images are created from UCIDv2 [35], an uncompressed image dataset, with a pair of first and second quality factors (QF1, QF2) = (a, b) where $a \neq b$. The quality factors considered in this chapter are 50, 60, 70, 80, and 90. Over the generated dataset, for both schemes, classifiers are independently trained and tested for 20 times and the average detection accuracies are reported. Per iteration, SVM [36] with random data partition (five-sixths for training and one-sixth for testing) has been employed to re-evaluate MP-162 on the generated dataset, the results of which are tabulated the result in Table 4.1. To re-evaluate MBFDF, FLD with randomly selected 1138 images for training and the 200 images for testing has been employed for 20 independent runs, the results of which appear in Table 4.2. Note that the re-evaluated detection rates slightly differ from what have been reported in [56] and [57] potentially because of: 1) different dataset, different random partition, and different features, for MP-162; 2) different random partition, for MBFDF.

Table 4.1 Detection Accuracies of MP-162

QF1/QF2	50	60	70	80	90
50	-	100.00%	100.00%	100.00%	100.00%
60	99.93%	-	100.00%	100.00%	100.00%
70	99.98%	100.00%	-	100.00%	100.00%
80	99.91%	99.64%	100.00%	-	100.00%
90	99.98%	99.96%	99.62%	99.62%	-

Table 4.2 Detection Accuracies of MBFDF

QF1/QF2	50	60	70	80	90
50	-	100.00%	100.00%	100.00%	100.00%
60	100.00%	-	100.00%	100.00%	100.00%
70	100.00%	100.00%	-	100.00%	100.00%
80	99.93%	99.63%	100.00%	-	100.00%
90	99.88%	99.93%	99.98%	99.95%	-

4.3 Shrink-and-Zoom Attack (SAZ)

Obfuscating DQ artifacts is central to anti-forensics. Such deliberate attacks have created credibility gap on many forensic schemes that rely upon the trace of DQ artifacts, for example, image tampering detection schemes derived from the DCT domain; however, in this anti-forensic work, the attention is limited to misleading double compression detection schemes of interest in the belief that the generalization of the attacks could be extended to other related forensic applications.

In this chapter, a simple attack on doubly compressed images which effectively fools double JPEG compression detection schemes is proposed. The rationale behind the attacks is to disrupt JPEG grid structure. The original concept is from steganalysis [58] in which the JPEG grid structure of a given JPEG image has been destroyed by calibration attack which follows the following procedures: 1) decompress a given JPEG image ; 2) crop four rows and four columns; 3) re-compress the cropped image with the same QF previously used. Although it has been proven that calibration can effectively reduce DQ artifacts, the inconsistency in image size caused by image cropping would leave an observable trail of image modification. In this dissertation, shrink-and-zoom (SAZ) attack is proposed on a doubly compressed image to suppress DQ artifacts while to maintain the

original image size as well as to preserve good image quality. The procedure of SAZ attack on a given doubly compressed image is as follows:

1. JPEG decompressing a given doubly compressed image.
2. Shrinking the image of size X by Y to sX by sY, where s is the degree of shrinkage and $0<s<1$.
3. Zooming the shrunk image back to its original size X by Y. In this work s = 0.9 and bilinear interpolation are recommended.
4. JPEG compressing the resultant image in Step 3 with QF2.

The strategy for disrupting JPEG grid structure employed in SAZ attack is basically double image resizing. Image resizing involves interpolation to estimate pixel values of the location previously non-existent. Even though there are many interpolation algorithms, bilinear interpolation is chosen because it not only retains reasonably good image quality but also powerfully attacks double compression detection schemes. Although image resizing has been mentioned in [59] as an alternative way to destroy JPEG grid structure, to the best knowledge, there is no anti-forensic scheme reported that this technique can fail double JPEG compression detection schemes.

It has experimentally been observed that the degree of shrinkage(s) and choice of interpolation have a strong connection with the effectiveness of DQ artifact elimination and image quality. Generally speaking, the smaller the s is, the worse the quality of anti-forensically modified image and the more severe the attack become; however, beyond some certain s, the effectiveness of the attack improves little while the image quality of the attacked image degrades greatly.

4.3.1 Image Quality Measurement (PSNR)

One of the ultimate goals of anti-forensic schemes is to preserve the visual quality of the image to be attacked. Usually expressed in terms of logarithmic decibel, peak signal-to-noise ratio (PSNR) has been widely used as a measure on image quality of reconstruction of lossy image and is defined as a function of mean squared error (MSE) of two $m \times n$ gray-scale images I and J. Either I or J is a noisy approximation of the other. Note that in the experimental setting presented in this chapter, anti-forensically modified images is considered as a noisy approximation of doubly compressed images which is considered as a reference image.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - J(i,j)]^2 \tag{4.1}$$

$$PSNR = 10 \log_{10} \left(\frac{Max_I^2}{MSE} \right) = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \tag{4.2}$$

Max_I is the maximum pixel value of the image, e.g., in this chapter, eight bits are used to represent one pixel, Max_I is 255.

Figure 4.1 Comparison between doubly compressed image and its SAZ attacked image. (left) Doubly compressed image generated from ucid01248 using (QF1, QF2) = (60, 80); (right) The anti-forensically modified image generated from SAZ with s = 0.9 and bilinear interpolation. The PSNR (dB) between two images is 35.97 dB (the left picture was used as a reference image).

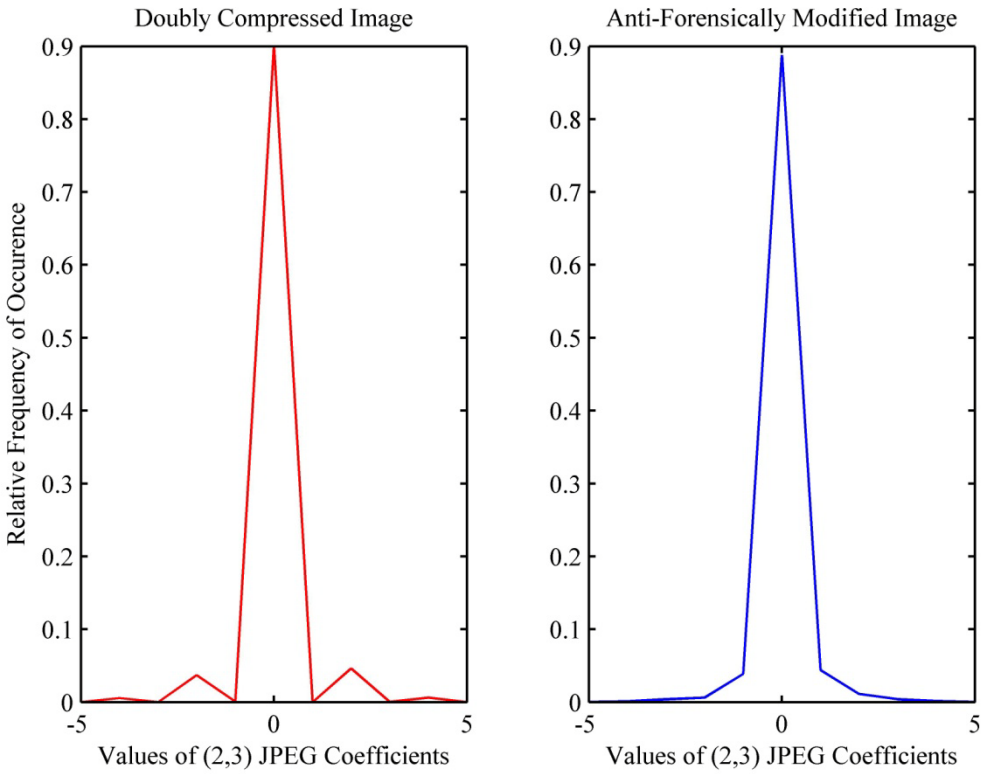

Figure 4.2 Mode histograms generated from the two images in Figure 4.1.

Throughout this chapter, the image ucid01248 in the UCID [35] dataset is used in the demonstrations of the proposed anti-forensic techniques which depict the

effectiveness of the techniques in terms of visual quality preservation and DQ artifact elimination capability. In Figure 4.1, not only the visual difference between the two images in the figure is virtually indiscernible subjectively but also the quantitative difference measure between the two images is quite acceptable. The histogram in Figure 4.2 (left) has a peak-and-valley pattern (DQ artifacts) caused by double JPEG compression while Figure 4.2 (right) contains no such a DQ artifacts associated with double JPEG compression which indicate that SAZ suppresses DQ artifacts to a considerable degree at a given pair of quality factors. Consequently, SAZ is capable of destroying DQ artifacts while preserving decent visual quality of the resultant anti-forensically modified images.

4.3.2 Evaluation on Anti-Forensic Scheme

In double JPEG compression detection, doubly compressed images are considered positive instances, while singly compressed ones are considered as negative instances. True positive (TP) rate is the percentage of doubly compressed images correctly classified, while true negative (TN) rate is the percentage of singly compressed images correctly classified. To evaluate the effectiveness of the proposed anti-forensic scheme, the key measure is the rate at which a classifier, trained for detecting double JPEG compression, classifies the anti-forensically modified images as doubly compressed images; the lower the more powerful of the proposed anti-forensics scheme, The evaluation process can be briefly described as follows: 1) conduct SAZ attack on a given doubly compressed image; 2) extract image feature vector; 3) feed it to the corresponding 20 trained classifiers. Table 4.3 shows the relationship between *s*, average PSNR, TP rates after SAZ attack on doubly compressed images generated from the UCID [35].

Table 4.3 Average PSNR, TP Rate after SAZ Attack versus *s*

s	0.998	0.9	0.5
PSNR (dB)	35.21	32.38	28.58
TP Rate (MP-162)	48.40%	33.74%	21.90%
TP Rate (MBFDF)	19.50%	12.40%	8.60%

Note that the average PSNR in dB is calculated by averaging the PSNR before converting it to dB. Table 4.3 reveals that smaller *s* brings about more effective attack while decreases the average PSNR (dB). To maintain acceptable image quality, $s = 0.9$ is chosen.

Table 4.4 True Positive Rates of MP-162

QF1\QF2	50	60	70	80	90
50	-	100.00%	100.00%	100.00%	100.00%
60	99.87%	-	100.00%	100.00%	100.00%
70	99.87%	100.00%	-	100.00%	100.00%
80	100.00%	99.60%	100.00%	-	100.00%
90	99.96%	99.96%	99.42%	99.64%	-

Table 4.5 True Positive Rates of MP-162 after SAZ Attack with $s = 0.9$

QF1\QF2	50	60	70	80	90
50	-	0.13%	0.00%	0.00%	0.00%
60	1.26%	-	0.00%	0.00%	0.00%
70	30.09%	19.10%	-	0.13%	0.00%
80	33.77%	38.21%	6.05%	-	0.13%
90	100.00%	99.78%	99.55%	9.87%	-

The average TP rates before SAZ attack are reported in Tables 4.4 and 4.6 in comparison with the corresponding average TP rates after SAZ attack in Tables 4.5 and 4.7. From Tables 4.4 to 4.7, it is noticeable that SAZ is more effective to mislead the trained classified when QF1<QF2 than when QF1>QF2. The intuitive explanation of this

phenomenon is discussed in Section 4.3.3 along with a quantitative measure to support the claim.

The effectiveness of SAZ is confirmed by the low TP rates after SAZ attack as shown in Tables 4.5 and 4.7. In all cases, SAZ almost perfectly attacks the upper triangle of the tables where QF1 < QF2; however, in the lower triangle of the tables where QF1 > QF2, SAZ is less effective. This phenomenon can be intuitively explained by the following reasons: 1) when QF1 < QF2, DQ artifacts are so obvious that SAZ can powerfully distort the statistical properties of the doubly compressed images; 2) when QF1 > QF2, DQ artifacts exist less severely, hence being more difficult for SAZ to distort the statistical properties of such doubly compressed images.

Table 4.6 True Positive Rates of MBFDF

QF1\QF2	50	60	70	80	90
50	-	100.00%	100.00%	100.00%	100.00%
60	100.00%	-	100.00%	100.00%	100.00%
70	100.00%	100.00%	-	100.00%	100.00%
80	99.85%	99.40%	100.00%	-	100.00%
90	99.75%	99.85%	99.95%	99.90%	-

Table 4.7 True Positive Rates of MBFDF after SAZ Attack with $s = 0.9$

QF1\QF2	50	60	70	80	90
50	-	1.00%	0.00%	0.00%	0.00%
60	10.55%	-	0.20%	0.00%	0.00%
70	30.80%	17.95%	-	0.00%	0.00%
80	1.60%	41.20%	10.00%	-	0.00%
90	90.65%	26.80%	14.80%	3.20%	-

4.3.3 Statistical Deviation by SAZ

The statistical changes in feature level introduced by SAZ are measured by comparing two statistical parameters *K* and *L* whose calculation is depicted in Figure 4.3 and values are tabulated in Tables 4.8 to 4.11. To calculate *K* and *L*, a singly compressed image with QF2 is used as a reference image. *K* is the L1 distance between feature vectors extracted from a given doubly compressed image and the reference image, while *L* is the L1 distance between feature vectors extracted from the anti-forensically modified image and the reference image. That is, *K* represents the change with respect to the reference image in feature-level introduced by double compression, *K* introducing the change with respect to the reference image in feature-level introduced by SAZ. Mean values of *K* and *L* are tabulated in Tables 4.8 to 4.11.

When QF1 < QF2, *L* is much less than *K* which means that SAZ moves the statistical properties of the anti-forensically modified images closer to their corresponding singly compressed versions (the reference images). When QF1 > QF2, *L* is generally close to *K* which means that SAZ slightly distort the statistical properties of the anti-forensically modified images. SAZ is therefore generally less effective under this situation. The amount of difference between *K* and *L* indicates the degree of statistical deviation introduced by SAZ.

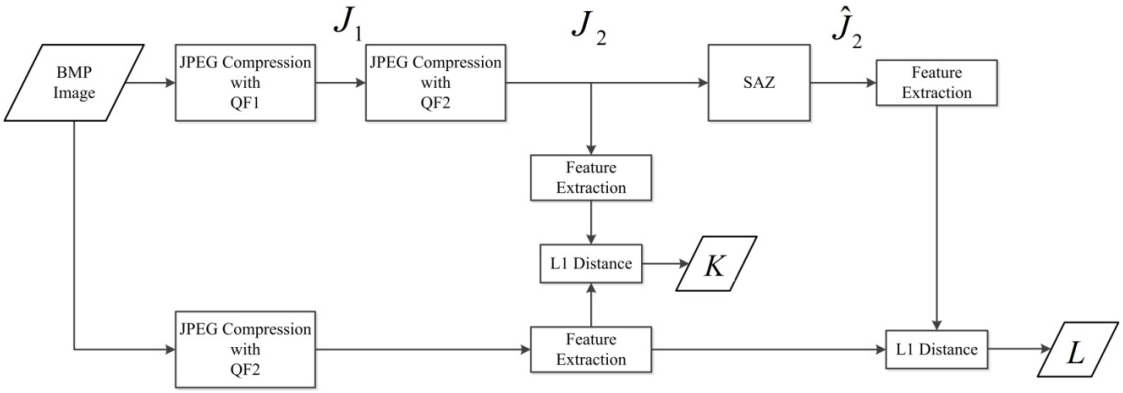

Figure 4.3 Block diagram of statistical deviation measure.

Table 4.8 Means of *K* of MP-162

QF1\QF2	50	60	70	80	90
50	-	2.97	11.14	13.26	18.26
60	1.82	-	5.39	14.85	16.05
70	1.79	1.57	-	4.40	15.06
80	1.45	1.24	1.89	-	11.82
90	0.30	0.44	0.36	1.02	-

Table 4.9 Means of *L* of MP-162

QF1\QF2	50	60	70	80	90
50	-	1.98	2.97	3.80	4.73
60	1.41	-	2.40	2.95	4.41
70	1.97	1.54	-	2.36	3.98
80	1.59	1.44	1.61	-	3.03
90	1.27	1.35	1.51	1.73	-

Table 4.10 Means of *K* of MBFDF

QF1\QF2	50	60	70	80	90
50	-	5.44	21.57	23.42	21.62
60	3.59	-	7.61	22.20	22.71
70	3.10	4.30	-	11.18	21.14
80	2.79	1.52	4.08	-	18.08
90	0.80	1.14	1.26	1.59	-

Table 4.11 Means of *L* of MBFDF

QF1\QF2	50	60	70	80	90
50	-	3.49	4.96	4.86	4.57
60	4.03	-	3.01	4.07	4.54
70	4.76	4.47	-	3.45	4.31
80	3.20	3.95	4.35	-	3.17
90	3.40	3.57	3.37	3.39	-

4.4 Alternatives to SAZ

As mentioned in Section 4.3.1, the ultimate goal of the anti-forensic scheme proposed in this dissertation is to disrupt JPEG grid structure. It has been shown in Sections 4.3.2 to 4.3.3 that SAZ, based on image resizing, is one of effective methods; however, it is beneficial to point out, although without large-scale empirical validation, a few other methods that could possibly be also effective in JPEG grid disruption.

First, image rotation is able to disrupt JPEG grid structure as also having been pointed out in [59]; however, this operation is not recommended because of the following reasons: 1) image rotation would require image cropping to eliminate artificial image boundary and result in the inconsistency between image sizes; 2) image rotation would make the spatial coordinate of the anti-forensically modified image nonaligned with the corresponding doubly compressed image which leads to unacceptably low PSNR.

Second, low-pass filtering can be considered as an equivalent operation of SAZ. There are of course various ways to apply low-pass filter to an image (pixel values outside the image boundaries are zero-order-hold interpolated); for the sake of simplicity, a 3×3 mean filter is chosen to apply to a given doubly compressed image and then JPEG compress the filtered image with QF2.

Figure 4.4 Comparison between doubly compressed image and its low-pass filtered image. (left) Doubly compressed image generated from ucid01248 using (QF1, QF2) = (60, 80); (right) The anti-forensically modified image generated by convoluting a 3×3 mean filter on the doubly compressed image. PSNR (dB) between two images is 34.57 dB (the left picture was used as a reference image).

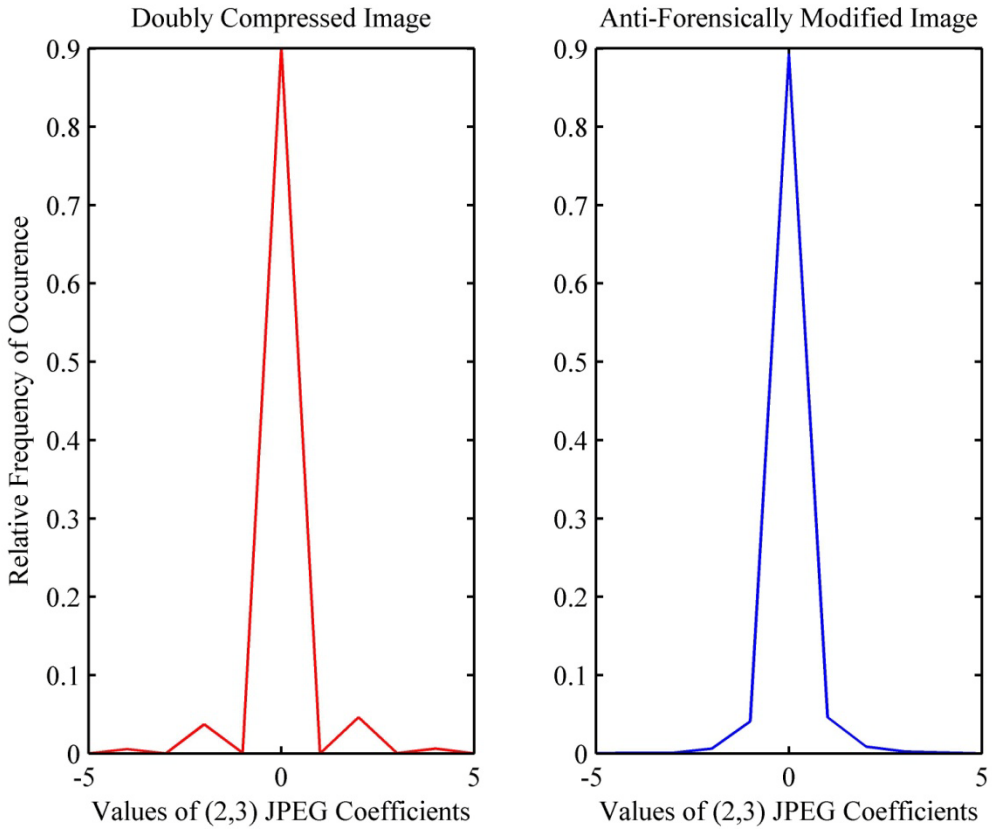

Figure 4.5 Mode histograms generated from the two images in Figure 4.4.

Figures 4.4 and 4.5 confirm the effectiveness of the usage of a low-pass filter not only to preserve image quality but also to eliminate DQ artifacts; under the same circumstances, the resultant PSNR suggests that image quality retained by convoluting a 3×3 mean filter to the corresponding doubly compressed image is slightly worse than that retained by SAZ. It is visually noticeable that the obtained anti-forensically modified image in Figure 4.4 (right) is a little more blurry than that in Figure 4.1 (left). This is, however, in no ways, a definite conclusion as other choices of low-pass filter may bring off a higher PSNR.

Third, intensity transformations, such as histogram equalization and logarithmic transformation, can also destroy DQ artifact. Such transformations globally change the relationship among pixel intensity in the given image; in some sense, depending on the degree of change introduced to a compressed image, such operations create a seemingly new uncompressed image with similar visual semantic content with virtually no DQ artifacts. In Figures 4.6 and 4.7, the effectiveness of applying histogram equalization to the given doubly compressed image is demonstrated. As expected, PSNR of the two images in Figure 4.6 is unacceptably low, only 10.63 dB, but DQ artifacts have been removed to some extent as shown in Figure 4.7. Although intensity transformation may work very effectively in hiding the traces of DQ artifacts, this approach is not recommended as it is most likely to yield a noticeably change in the attacked images.

Figure 4.6 Comparison between doubly compressed image and its histogram-equalized image. (left) Doubly compressed image generated from ucid01248 using (QF1, QF2) = (60, 80); (right) The anti-forensically modified image generated by conducting histogram equalization. PSNR (dB) between two images is 10.63 dB (the left picture was used as a reference image).

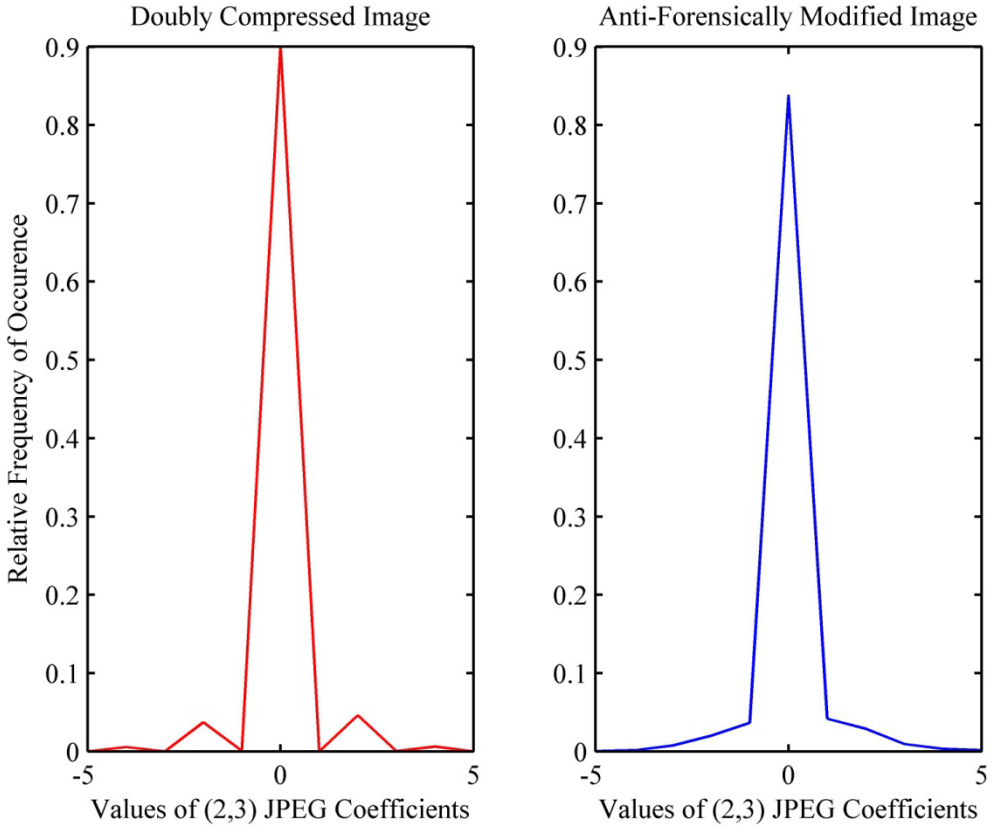

Figure 4.7 Mode histograms generated from the two images in Figure 4.6.

4.5 Discussion and Conclusions

4.5.1 SAZ in a Nutshell

In this chapter, an anti-forensic operation capable of misleading two highly effective JPEG double compression detection schemes [56] and [57] is proposed. There are two ultimate goals of anti-forensic schemes: 1) image quality preservation; 2) obfuscating forensic artifacts. The proposed attack relies upon image resizing with bilinear interpolation and is called shrink-and-zoom (SAZ). The efficacy of SAZ is assessed by the rate at which a classifier, trained for double JPEG compression detection, classifies anti-forensically modified images as doubly compressed images. That is, the lower such a rate, the more powerful the attack is. To operate SAZ, the scaling factor s is the key parameter that controls not only the image quality but also TP rates after the attack. s = 0.9 which yields a reasonable balance between PSNR and TP rates after the attack.

The performance gap between TP rates between before and after SAZ attack also indicates how effective the attack is; that is, the lower TP rate after the attack than TP rate before the attack, the more effective the attack is. Although in most cases, the TP rates after the attack are lower than those before the attack, it has been observed that TP rate after the attack is slightly higher than TP rate before the attack for only two combinations of QF1 and QF2 and in such cases the attack can be deemed ineffective.

To explain the situations in which SAZ brings forth a lower TP rate after the attack than before the attack and those in the opposite situations, PCA 2D projections of some representative cases are depicted in Figures 4.8 and 4.9: 1) MP-162 features of (QF1, QF2) = (60,80) of which TP rate after the attack = 0% while TP rate before the

attack = 100%; 2) MP-162 features of (QF1, QF2) = (90, 50) of which TP rate after the attack = 100% and TP rate before the attack = 99.96%.

In Figure 4.8, the scatter plot of the projection of MP-162 features extracted from corresponding anti-forensically modified images is very much closer to that from singly compressed images; that's, in feature space, SAZ moves the statistical properties of anti-forensically modified images much closer to those of singly compressed images. On the other hand, in Figure 4.9, the scatter plot of the projection of MP-162 features extracted from corresponding anti-forensically modified images is closer to that from doubly compressed; in more details, SAZ, under this circumstance, moves the statistical properties of anti-forensically modified images further away from those of singly compressed images. The relationship between TP rates before and after the attack implies the direction toward which SAZ moves the statistical properties of features, while the difference between *K* and *L* define the degree of statistical deviation SAZ brings forth.

4.5.2 Connection between Anti-Forensic Schemes of Double JPEG Compression Detection and the Practicality of Image Tampering Detection Schemes

Image tampering detection has been a hot research topic over the past few years. Many such schemes, such as [20], [12], and [13], have been proposed to measure irregularity caused by image tampering from observed JPEG coefficients and evaluated over some public datasets, e.g., [27] and [28]. These schemes generally perform fairly well within the datasets used; however, its practicality is still questionable as it has been reported in [13] that the accuracy of testing real-life tampered images, of which the ground truths are known but the processing histories done to the images are not, collected from the Internet

on the classifiers trained over [28] is much lower than that evaluated by using the testing images in the used dataset.

What has been presented all along this chapter could partly explain why image tampering detection schemes derived from JPEG coefficients have performed poorly in reality. Assuming no further malicious attack, image resizing and image rotation seem to be frequent choices of non-malicious image processing operation that makes a given image suitable for being displayed in an Internet browser while image enhancement operations like histogram equalization as well as low-pass seem to be done relatively much less. However, any of such non-malicious image operations is likely to be applied to any digital image before its distribution over the Internet and it can potentially obfuscate the traces of tampering artifacts left in the JPEG coefficients. In other words, for compressed domain methods, tampered images generated from JPEG images are susceptible to many common image processing operations; consequently, the practicality of such methods is limited. SAZ as well as some other methods discussed in Section 4.4 is most likely to be able to confuse image tampering detection schemes that rely on JPEG information.

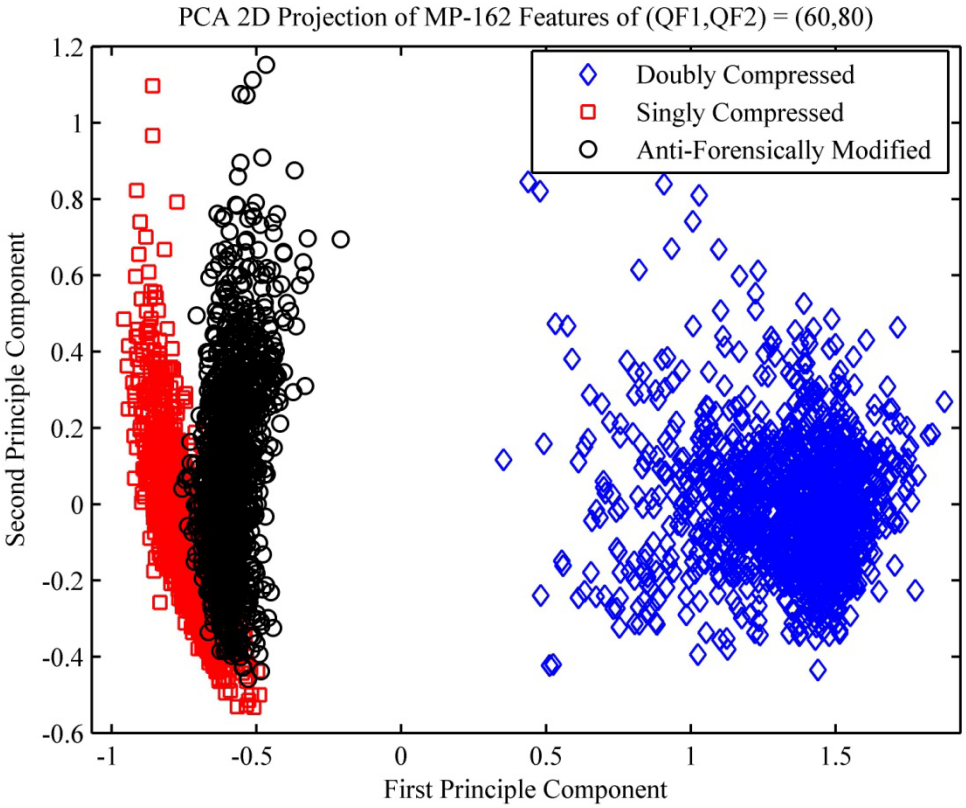

Figure 4.8 PCA 2D projection of MP-162 features of (QF1,QF2) = (60,80).

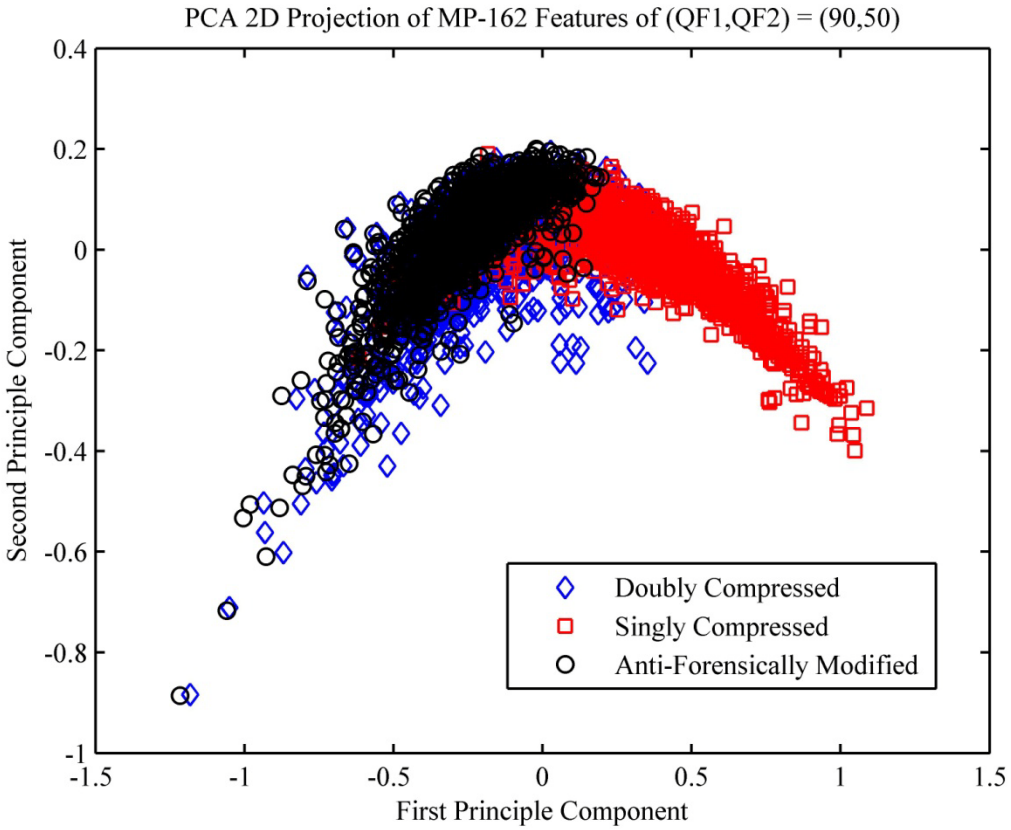

Figure 4.9 PCA 2D projection of MP-162 features of (QF1,QF2) = (90,50).

CHAPTER 5

TEXTURAL FEATURES FOR STEGANALYSIS

It is observed that the co-occurrence matrix, one kind of textural features proposed by proposed by Haralick et al. [60], has played a very critical role in steganalysis. On the other hand, the data hidden in the image texture area has been known difficult to detect for years, and the modern steganographic schemes tend to embed data into complicated texture area where the statistical modeling becomes difficult. Based on these observations, in this chapter, the textural features from the rich literature in the field of texture classification are studied for further development of the modern steganalysis. As a demonstration, a group of textural features, including the local binary patterns, Markov neighborhoods and cliques, and Laws' masks, have been selected to form a new set of 19,593 features, which are used with the FLD-based ensemble classifier to steganalyze the BOSSbase. The related work is to be published in [61].

5.1 Introduction

Steganography and steganalysis are a pair of modern technologies that have been moving ahead swiftly in the last decade. The conflicting between these two sides is a driving force for the rapid development. That is, each side learns from its counterpart. From the modern steganalysis point of view, Avcibas et al. [62] introduced the machine learning framework, consisting of statistical features and classifier for the first time. Lyu and Farid [63] proposed the first four statistical moments of wavelet coefficients and their prediction errors of nine high frequency sub-bands from three-level decomposition are

used to form a 72-D feature vector for SVM classification for steganalysis. Harmsen [64] proposed the steganalysis method based on the mass center of histogram characteristic function has shown improved effectiveness in steganalysis. Xuan et al. [65] utilized a framework combining wavelet decomposition and moments of characteristic functions. To break steganographic schemes with popularly used JPEG images as carriers, such as OutGuess, F5 and Model-based steganographic schemes, Fridrich [59] used a group of 23 features, including both the first and second order statistics, have been used together with a calibrate technique. Sullivan et al. [66] firstly utilized Markov process for steganalysis. How to handle the high dimensionality of elements in the transition probability matrix resultant from the application of Markov process has been studied in Zou et al. [67], for the spatial domain and in [32] for the JPEG domain (frequency domain). In Pevny et al. [68], both the first and the second order Markov models, called SPAM, have been established to detect the more advanced steganographic scheme known as LSB matching. As expected, there is no end in the competition between steganography and steganalysis just like mouse versus cat. A modern steganographic scheme, named HUGO (Highly Undetectable Steganography) [7], has been developed so as to fail the SPAM by taking high order difference into consideration in its data embedding. Steganalytic methods [69-71] have been reported to break HUGO. In Gul and Kurugollu [71], image features are extracted via applying high-pass filters to the image, followed by down-sampling, feature selection, and some optimization technique. Depending on the chosen parameters, the feature dimensionalities range from more than one hundred to more than one thousand; with a linear classifier, the detection accuracies of the generated features on BOSSbase 0.92 [72, 73] (the dataset established for the HUGO competition.) range from 70% to

more than 80%. In Fridrich el al. [69, 70], the difference arrays from the first-order up to the sixth-order are all used for feature extraction in addition to other newly designed features, resulting in the total number of features as high as 33,963. Because of the high feature dimensionality, an ensemble classifier using Fisher's Linear Discriminant (FLD) has been developed and utilized. These novel measures result a detection rate of 83.9% on BOSSbase 0.92 [72, 73] at the embedding rate of 0.4 bits per pixel (bpp).

What described above is by no means a complete review of this active research field in steganalysis. For instance, the recent technologies of steganography and steganalysis in the JPEG domain have not been discussed here, which however have shown the same pattern of competition among these two areas. The observation from the above-discussion is that the modern steganalysis has made rapid progress in the past decade, so does modern steganography.

5.1.1 Image Dataset and Classifiers

Throughout this chapter, the image dataset BOSSbase 0.92 [72, 73], which is the standard dataset for HUGO steganalysis evaluation, will be used for all the experiments including empirical validations. The dataset consists of 9,074 gray-scale uncompressed images. An example of a pair of stego and cover images is shown in Figure 5.1. The stego images are generated by HUGO at 0.4 bpp with default parameters. Features extracted from the dataset are evaluated in the same random partition. That is, for each training and testing phase, randomly selected 8,094 patterns are for training and the 1,000 left for testing.

For some small-scale empirical validation, FLD is employed; however, for some large-scale empirical validations, FLD-based ensemble classifiers [69, 70] is used instead. For the ensemble classifiers, d_{red} (the dimensionality of random selected feature

sub-set) and L (the number of base learners) are involved in the determination of final accuracy which is computed based on majority voting rule. A proper choice of the two parameters can be obtained by brute force searching. In this chapter, $d_{red} \in \{1{,}800\ 2{,}400\ 2{,}800\}$ and $L \in \{1, 2, \ldots, 101\}$.

The final FLD-based ensemble accuracy of the whole feature set, including some small-scale empirical validations, is evaluated over 12 independent runs of classification while some large-scale empirical validations over three independent runs.

Figure 5.1 Example of stego (left) and cover (right) images [72, 73].

5.1.2 Milestones of Textural Features

In this chapter, a different look at steganalysis is taken from the texture classification point of view. Texture classification is one of the major applications in texture analysis which has a long rich history. There are a variety of statistical methods in texture classification, some of which have been successfully used in the early days of steganalysis. Among such methods, histogram and its moments are considered the first-order statistics in which no spatial relationship among pixels is taken into account.

According to the highly cited (as of February 2012, having been cited almost 7,000 times according to Google) paper by Haralick et al. [60] in 1973, "context, texture, and tone are always present in the image, although at times one property can dominate the other," "texture is an innate property of virtually all surfaces." In [60], the co-occurrence matrix, the second-order statistics in which the spatial relationship between two pixels is considered, has been proposed as textural features for image classification. Since then it has been one of the most widely used statistical methods for various tasks in pattern recognition.

Now the thought in [60] is further extended. The modern steganography hides data into a cover image. That means the original texture of cover image has been modified somehow after data embedding even though the change is small. Therefore many technologies developed for texture images classification are reasonably expected to be usable for steganalysis. In addition, it has been reported that the data hidden inside the texture images are difficult to be detected, for instance, by Bohme [74], in other words, the texture images are suitable for steganography, consequently the steganalysis on texture images is challenging, and some efforts have been made, for instance, Chen el al. [75]. Therefore, it becomes clear that the technologies developed for texture images classification should be able to play an important role in modern steganalysis.

In 1976, Weszka et al. [76] presented a comparative study on the performance of three major kinds of textural features in terrain classification. The study reveals that the first-order statistics of the absolute gray difference performs comparably with co-occurrence matrix. Both of the afore-mentioned methods outperform the Fourier power spectrum. In 1992, Ohanian and Dubes [77] conducted a comparative study on the

performance of four types of features in texture classification: Markov Random Field parameters, multi-channel filtering features (wavelet transform), fractal-based features, and co-occurrence features. The study shows that the co-occurrence features perform the best among the four types of features. That is why co-occurrence matrix has been an attractive choice for texture classification and related applications.

In fact, there are many tools developed for texture classification that can be borrowed to use for steganalysts in addition to co-occurrence matrix (transition probability matrix can be shown equivalent to the co-occurrence matrix under certain condition which has been used in steganalysis). Specifically, by taking a close look at the techniques used in texture classification, for instance, according to [78], Markov random fields (MRF) and others which belong to the technologies suitable for stationary texture images may be useful for steganalysis. In the category of non-stationary texture images, there are Laws' masks, local binary patterns (LBP), and others.

These thoughts are the motivation to investigate new steganalysis technologies. First the LBP technologies [79, 80], respectively proposed in 1996 and 2002, is examined. In this popular technology (as of February 2012 [80] has been cited almost 1900 times according to Google), the pixels in the entire image (or in the area of interesting) are examines. For each considered pixel, the LBP opens, say, a 3×3 neighborhood surrounding it. Then the gray-value of each of the eight neighbor pixels is compared with that of the central pixel. If the gray-value of a neighbor pixel is smaller than that of the central pixel, a binary zero is recorded for this pixel; otherwise, a binary one is recorded; thus resulting a string of eight binary bits, each being either zero or one. This procedure is conducted for each pixel of the given image. If one chooses a

sequencing among these eight binary bits assigned to the eight-neighbors, one then obtain a corresponding eight-bit binary number. Applying this procedure to all pixels, eight-bit binary numbers can be derived, specifically one for each pixel of the image under consideration (pixel values outside the image boundaries are zero-order-hold interpolated). Since any eight-bit binary number corresponds to a specific decimal number in a range from zero to 255, clearly, the histogram of all of the decimal numbers thus formulated consists of 256 bins. The distribution of this type of histogram bins' values is chosen to characterize the given image. Since it is obtained from each individual pixel through comparing it with its local neighbor pixels, this type of histogram is expected to be suitable for texture classification; in this case, for steganalysis.

Note that there are several different ways to generate the histogram. A popular way of LBP technology used in texture analysis ends up with only 59 bins for the 3×3 neighborhoods described above. That is, the statistics shows that there are many very sparse bins among the 256 bins. Some of these 256 bins are merged so as to result in only 59 bins without losing much information in classification. In order to achieve rotation invariance, the following procedures are taken. That is, a unit circle is considered from the central pixel with a radius being one; hence, the gray values of four corner pixels of this 3×3 block are determined by interpolation.

Furthermore, the LBP technology can be employed in a multi-resolution manner. That is, in addition to a neighborhood of 3×3, one can also consider neighborhood of 5×5 and/or 7×7. It is shown in [80] that multi-resolution does help in texture classification. In addition to the linear binary patterns just discussed, the LBP scheme also considers "contrast" by introducing another quantity called variance. That is, if for the case of 3×3

square neighborhoods, the mean average of the eight surrounding pixels' gray-value is first calculated, followed by the calculation of the local variance with respect to the central pixel's gray-value. For detail of the LBP technologies, readers are referred to [79, 80].

As an exercise, these textural features are applied to steganalyzing HUGO stego dataset [72, 73] designed for the BOSS contest. A steganalyzer with 19,593 features derived from the textural features is constructed. Instead of co-occurrence matrix, employed are LBP features (59-D, corresponding to the above mentioned 59 bins, used for some filtered 2-D array, and 256-D (256 bins) used for others) and variance features derived from the multi-resolution way. In addition, Laws mask and the mask and cliques associated with Markov Random fields [78] are utilized. The classifier utilized is the FLD-based ensemble classifier, reported in [69, 70]. The achieved average detection rate is 82.60% on BOSSbase 0.92 [72, 73]. While the first-stage work has been positive, more works need to be done to further move this investigation ahead. It is hope that a different angle to view and handle steganalysis has been opened by this study.

5.2 Image Statistical Measures

Advanced steganographic schemes such as HUGO [7] tend to embed data into cover image locally into some selected regions so as to make the image statistical modeling difficult, especially into highly texture regions. Intuitively, this small local change should be efficiently captured by some image operators which emphasize on modeling microstructure image properties. In this chapter, the local binary pattern (LBP) operators

[79, 80], which have been popularly used in texture classification arena, are introduced as a potential statistical image modeling for steganalysis.

5.2.1 Local Binary Pattern

Ojala et al. [79] proposed LBP to model the statistics of a texture unit defined within a neighborhood of, say, 3×3 pixels. Each of eight neighboring pixels of a 3×3 neighborhood is thresholded by the gray value of its central pixel to form an eight-bit binary pattern. Figure 5.2 depicts a 3×3 neighborhood employed in the calculation of the original LBP in which g_c is the center pixel and g_p, $p = 0,2,\ldots,P\text{-}1$, where P is the number of neighboring pixels and equal to eight in this case, representing the neighboring pixels. Generalized to different P values and correspondingly defined neighborhoods, Equations (5.1) and (5.2) express the formulation of LBP mathematically.

g_3	g_2	g_1
g_4	g_c	g_0
g_5	g_6	g_7

Figure 5.2 3×3 neighborhood.

$$LBP = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \tag{5.1}$$

$$s(x) = \begin{cases} 1, x \le 0 \\ 0, x > 0 \end{cases} \tag{5.2}$$

Consequently, a histogram of 256 bins is formulated as a texture descriptor which represents vital information about spatial structure of image texture at microscopic level. The basic LBP is denoted as LBP_8.

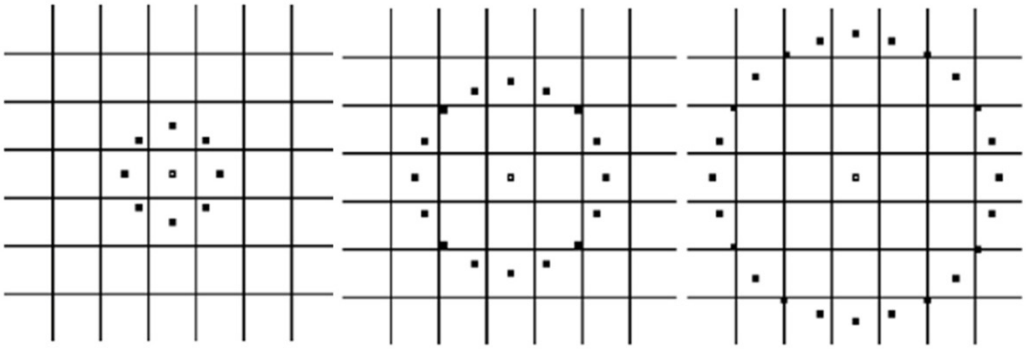

Figure 5.3 Circular symmetric neighbor sets (adapted from [80]).

In [80], Ojala et al. reported that LBP operators could achieve rotation invariant property after some manipulation. In this version of LBPs, the local neighborhood is circularly defined as shown in Figure 5.3 in which the pixel values of the neighbors falling outside the center of the pixel grids are estimated by interpolation. The computation of LBP values still follows Equations (5.1) and (5.2). Rotation invariant and uniformity mappings are introduced. The authors classify LBP into two categories as shown in Figure 5.4: "uniform" (top row) and "non-uniform" (bottom row) patterns. Uniform patterns have the number of binary transitions between zero (black dot) and one (white dot) over the whole neighborhood circle less than or at most equal to two while the patterns whose number of such transitions is greater than two are considered as non-uniform. In texture classification, uniform patterns often occupy the majority of the histogram which makes merging non-uniform patterns into the same bin legitimate. This pattern merging is simply called uniformity mapping (or u2 mapping), reducing the

number of bins in a histogram from 256 to 59 bins. This type of LBP descriptor is denoted as $LBP_{P,R}^{u2}$ where P defines the number of neighbor pixels, R the radius of the circular symmetric neighborhood and u2 uniformity mapping.

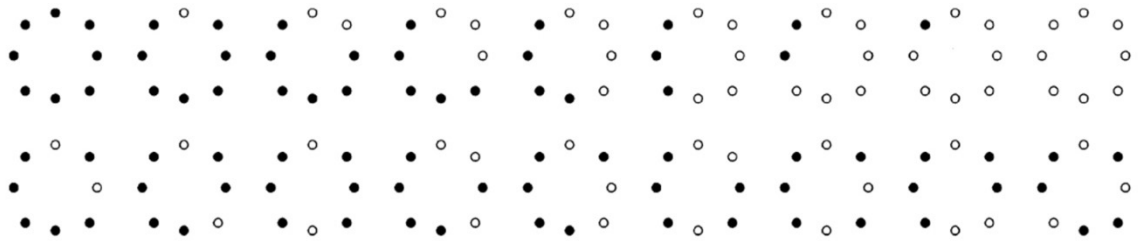

Figure 5.4 Examples of uniform patterns (all in the top row) and non-uniform patterns (all in the bottom row) (adapted from [80]).

The authors also suggested a feasibility of enhancing texture classification performance by incorporating multi-resolution approach.

Note that in this chapter only $P = 8$ is utilized in order to keep feature dimensionality manageable and that the circular symmetric neighbor inscribed within 3×3 square neighborhood when $R = 1$, 5×5 when $R = 2$, and 7×7 when $R = 3$.

5.2.2 Contrast Measure

In some applications, the performance of LBP can be enhanced by the use of a local contrast measure [80]. In this chapter, local contrast is measured in a 3×3 square neighborhood and as a result a variance image can thus be formed. The contrast measure on square 3×3 neighborhood is denoted as VAR_8 and is defined as follows.

$$VAR_8 = \tfrac{1}{8}\sum_{p=0}^{7}(g_p - \mu_8), \quad \text{where } \mu_8 = \tfrac{1}{8}\sum_{p=0}^{7} g_p \tag{5.3}$$

It has been observed empirically that LBP features extracted from some variance images can enhance the detectability of the proposed steganalyzer.

To demonstrate the effectiveness of LBP operators in steganalysis, some simple testing scenarios are constructed to compare the performance of features derived from LBP operators with those from co-occurrence matrix. Here a set of features on the first-order horizontal residual images generated by filtering images in BOSSbase 0.92 [72, 73] with the operator [-1 1] is used.

To derive feature using co-occurrence matrix along horizontal direction, the residual images are first thresholded with $T = 4$ which results in the feature dimensionality of 81 [67, 68]. The corresponding feature dimensionalities of LBP_8, $LBP_{8,1}^{u2}$ (i.e., as introduced, eight neighbor elements in total, radius being one, u2 mapping applied), $LBP_{8,2}^{u2}$ and $LBP_{8,3}^{u2}$ are 256, 59, 59, and 59, respectively. FLD is employed with 12 independent runs.

Table 5.1 Comparative Performance Study of Co-Occurrence and LBP Features from Horizontal Difference Array

Type	I	II	III	IV
TP Rate	57.48%	56.61%	64.53%	61.56%
TN Rate	51.46%	56.98%	65.20%	61.15%
AC	54.47%	56.80%	64.87%	61.36%
Dimensionality	81	59	256	177

In Table 5.1, type I stands for features derived by using co-occurrence matrix formulated along horizontal direction, II by $LBP_{8,1}^{u2}$, III by LBP_8 and IV by $LBP_{8,1}^{u2}$ $+LBP_{8,2}^{u2}$ $+LBP_{8,3}^{u2}$.

The comparative performance is shown in Table 5.1 where the statistics shows that: 1) features generated from LBP_8 are much more powerful than those from co-

occurrence matrix but with a higher dimensionality; 2) features generated from $LBP_{8,1}^{u2}$ perform slightly better than those from co-occurrence matrix although they are of lower dimensionality; 3) multi-resolution approach improves the performance of $LBP_{P,R}^{u2}$ scheme while it keeps dimensionality manageable.

Instead of using co-occurrence matrix, in this chapter statistical image features are formulated based solely on LBP operators. In so doing, an LBP operator is applied to a set of residual images, each of which reveals artifacts associated with steganography in a different way.

5.3 Content-Adaptive Prediction Error Images

Small perturbation to cover image caused by steganographic schemes may be considered as a high frequency additive noise; as a result, eliminating low-frequency representation of images before feature extraction process would make the resulting image features better represent the underlying statistical artifacts associated with steganography. With the modern steganographic schemes such as HUGO [7], it is intuitive that the prediction error images (also referred to as residual images) generated in a content-adaptive manner would effectively reveal such artifacts caused by data embedding. In this chapter, I is denoted as image, R as residual image, and $Pred(I)$ as corresponding predicted image. Predicted images here are calculated based on some relationship within a predefined square neighborhood. Mathematically, R can be expressed in Equation 5.4.

$$R = I - Pred(I) \tag{5.4}$$

In this subsection, the following two major kinds of content-adaptive residual images are proposed. The first kind is generated based on the proposed prediction scheme modified based on [81], while the second kind is generated based on a collection of median filters.

5.3.1 Local Successive Prediction Error Image

Shi et al. [82] utilized the predication scheme [21] in the proposed steganalyzer. A modified prediction scheme based on [21] is adopted in this chapter to better reveal steganographic artifacts utilizing a 3×3 neighborhood to formulate the prediction error. Since the application is not coding, it is free to manipulate the prediction scheme. That is, the prediction scheme [81] is employed in a 2×2 neighborhood but in a different way; that is, with a fixed reference pixel (a pixel to be predicted), the 2×2 neighborhood has been rotated for four times to cover a 3×3 neighborhood, each rotation yielding one predicted value of the reference pixel. The final predicted value is the average of these four predicted pixel values. Figure 5.5 and Equation 5.5 describe the prediction process. Note that the four-time rotations of the 2×2 neighborhood to form four 2×2 neighborhoods are the contribution of Licong Chen.

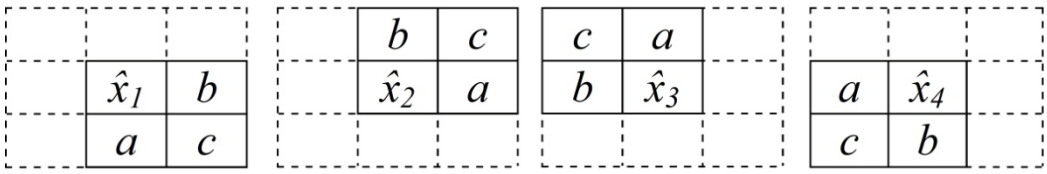

Figure 5.5 Four 2×2 neighborhoods used to predict the center pixel of a 3×3 neighborhood.

$$\hat{x}_i = \begin{cases} \max(a,b) & c \le \min(a,b) \\ \min(a,b) & c \ge \max(a,b) \\ a+b-c & \text{otherwise} \end{cases} \tag{5.5}$$

The empirical validations to be shown in Tables 5.2 are based on 12 independent runs of FLD over the dataset. Table 5.2 shows that features extracted from residual images generated by this proposed scheme in which *Pred(I)* = $(\hat{x}_1 + \hat{x}_2 + \hat{x}_3 + \hat{x}_4)/4$ are more discriminative than those used in [82] in which *Pred(I)* = $\hat{x}_1$.

Table 5.2 Performance Comparison of Features from Prediction Error Images

Prediction Scheme	Original [81]	Proposed
AC	60.18%	61.29%
Dimensionality	59	59
Descriptor	$LBP_{8,1}^{u2}$	$LBP_{8,1}^{u2}$

Much of image content has been removed by the proposed scheme; however, the influence of image content can be further reduced by successive application of this scheme. In this chapter, PE_n is denoted as a prediction error image generated by applying the proposed scheme to the original input image for n multiple times.

5.3.2 Median-Filter-Based Prediction Error Images

Spatial filters have been widely used as low-pass filters. Much of their applications are for image denoising. It is therefore intuitive to generate residual images by using median filters to compute predicted images. That is, a median filtered image is subtracted from an original image, thus generating a prediction error image. In this chapter, a set of median filters of three different sizes, 3×3, 5×5, and 7×7, is used to calculate predicted images. *Pred(I)* in Equation (5.4) is defined by the output of applying a median filter defined here

to a given input image *I*. Figure 5.6 depicts symbolic representations of pixel locations in three different sizes of square neighborhoods while Table 5.3 shows the pixel locations to be involved in the local median calculations.

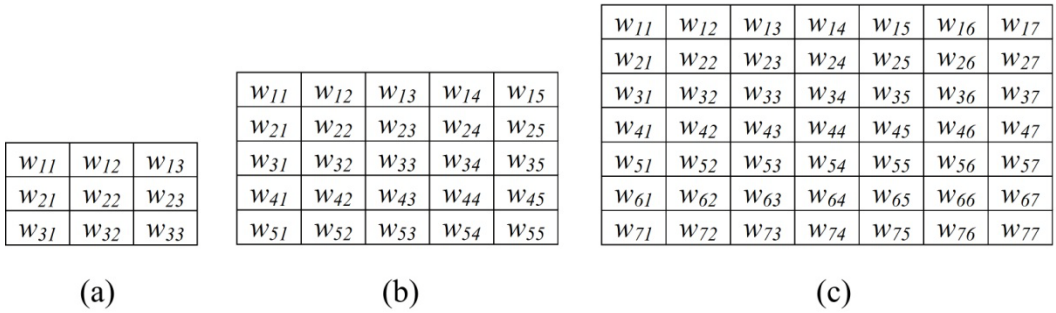

Figure 5.6 Symbolic representations of pixel locations used in the creation of median-filter-based prediction error images. (a) 3×3, (b) 5×5, and (c) 7×7 neighborhood.

Table 5.3 Configuration of Median Filters Employed in Generating Median-Filter-Based Prediction Error Images

Mask size	Filter number	Pixel locations used in computing median image
3×3	1	w_{11}, w_{13}, w_{22}, w_{31}, w_{33}
	2	w_{12}, w_{21}, w_{22}, w_{23}, w_{32}
5×5	1	w_{12}, w_{14}, w_{21}, w_{22}, w_{24}, w_{25}, w_{33}, w_{41}, w_{42}, w_{44}, w_{45}, w_{52}, w_{54}
	2	w_{11}, w_{13}, w_{15}, w_{31}, w_{33}, w_{35}, w_{51}, w_{53}, w_{55}
	3	w_{13}, w_{22}, w_{23}, w_{24}, w_{31}, w_{32}, w_{33}, w_{34}, w_{35}, w_{42}, w_{43}, w_{44}, w_{53}
7×7	1	w_{12}, w_{13}, w_{15}, w_{16}, w_{21}, w_{22}, w_{23}, w_{25}, w_{26}, w_{27}, w_{31}, w_{32}, w_{33}, w_{35} w_{36}, w_{37}, w_{44}, w_{51}, w_{52}, w_{53}, w_{55}, w_{56}, w_{57}, w_{61}, w_{62}, w_{63}, w_{65}, w_{66} w_{67}, w_{72}, w_{73}, w_{75}, w_{76}
	2	w_{14}, w_{22}, w_{24}, w_{26}, w_{34}, w_{41}, w_{42}, w_{43}, w_{44}, w_{45}, w_{46}, w_{47}, w_{54}, w_{62} w_{64}, w_{66}, w_{74}
	3	w_{11}, w_{13}, w_{15}, w_{16}, w_{31}, w_{33}, w_{35}, w_{37}, w_{44}, w_{51}, w_{53}, w_{55}, w_{57}, w_{71} w_{73}, w_{75}, w_{77}
	4	w_{14}, w_{23}, w_{24}, w_{25}, w_{32}, w_{33}, w_{34}, w_{35}, w_{36}, w_{41}, w_{42}, w_{43}, w_{44}, w_{45} w_{46}, w_{47}, w_{52}, w_{53}, w_{54}, w_{55}, w_{56}, w_{63}, w_{64}, w_{65}, w_{74}

5.4 Residual Images Based on Laws' Masks

It is always feasible to extract image features from a more favorable SNR image which is less influenced by image content. Note that stego-noise is the signal considered here. The residual images in this portion are computed by applying high-pass filters to the given image in the spatial domain. Some residual images in this part are also generated in a content-adaptive manner by incorporating two non-linear operators, minimum and maximum in order to catch the desired artifacts.

This part of image statistical features is formulated by two major set of 1-D spatial high-pass filters. The first set of high-pass filters is Laws' masks [18] which are of odd sizes (three, five, and seven), while the other set which contains even-tap high-pass filters (two, four, and six) have been adopted in this doctoral research.

As shown in Table 5.4, F4 and F6 are generated by convolving the mask [-1 1], popularly used in steganalysis and denoted by F2 in this chapter, with *S3* and *E5*, respectively, which are shown in Table 5.4.

Table 5.4 High-Pass Filters Employed in the Creation of Residual Images in Section 5.4

Category	Number of Taps	Name	Filter
Laws' Masks	3	Edge 3 (*E3*)	[-1 0 1]
		Spot 3 (*S3*)	[-1 2 -1]
	5	Edge 5 (*E5*)	[-1 -2 0 2 1]
		Spot 5 (*S5*)	[-1 0 2 0 -1]
		Wave 5 (*W5*)	[-1 2 0 -2 1]
		Ripple 5 (*R5*)	[1 -4 6 -4 -1]
	7	Edge 7 (*E7*)	[-1 -4 -5 0 5 4 1]
		Spot 7 (*S7*)	[-1 -2 1 4 1 -2 -1]
		Wave 7 (*W7*)	[-1 0 3 0 -3 0 1]
		Ripple 7 (*R7*)	[1 -2 -1 4 -1 -2 1]
		Oscillation 7 (*O7*)	[-1 6 -15 20 -15 6 -1]
Even Taps	2	Filter 2 (*F2*)	[-1 1]
	4	Filter 4 (*F4*)	[1 -3 3 -1]
	6	Filter 6 (*F6*)	[1 -3 2 2 -3 1]

For a given filter, five different residual images can be generated as follows: 1) R_h by applying a filter in the horizontal direction; 2) R_v by applying a filter in the vertical direction; 3) R_{hv} by applying a filter in the horizontal direction and then in the vertical direction in a cascaded manner; 4) $R_{min} = \min(R_h, R_v, R_{hv})$; 5) $R_{max} = \max(R_h, R_v, R_{hv})$.

5.5 Residual Images Based on Markov Neighborhoods and Cliques

Markov Random Field (MRF) has been widely used in texture classification, segmentation and texture defect detection [78]. Based on Markov condition, a neighborhood, called Markov neighborhood, can be constructed, into which the Markov parameters can be assigned as weights. These neighborhoods are characterized by a group of pixels with a variety of orientations often symmetrically inscribed within a square window of odd size. They are hence tempting choices for advanced steganalysis. In this doctoral research, the immediate application of Markov neighborhood is for high-pass filtering instead of texture classification. As a result, there is no need to strictly rely on Markov condition and parameters. Figure 5.7 represents the masks used to generate residual images described in this portion. Note that the mask in Figure 5.7 (e) is derived from Figure 5.7 (d) in this chapter.

In addition to Markov neighborhoods, cliques, portions of Markov neighbors, are proposed to use as high-pass filters. Cliques used in this research work are shown in Figure 5.8. The artifacts caused by steganalysis, reflected in residual images and obtained by applying these cliques are more localized than those caught by applying Markov neighborhood because of their small sizes. Thus, the detectability of the proposed steganalysis scheme has been enhanced.

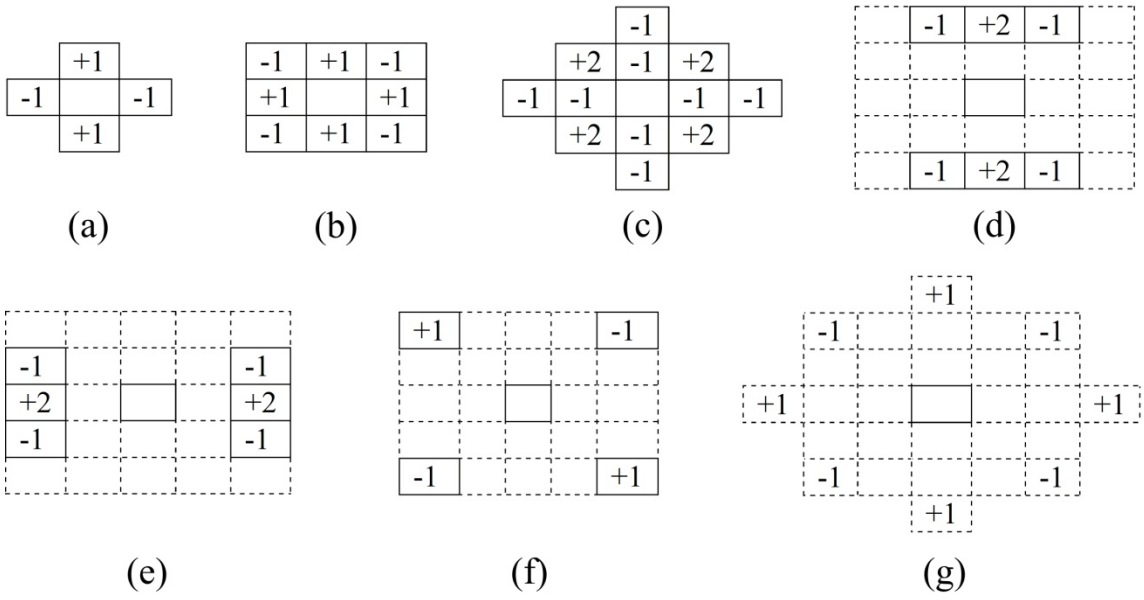

Figure 5.7 High-pass filters based on Markov neighborhoods.

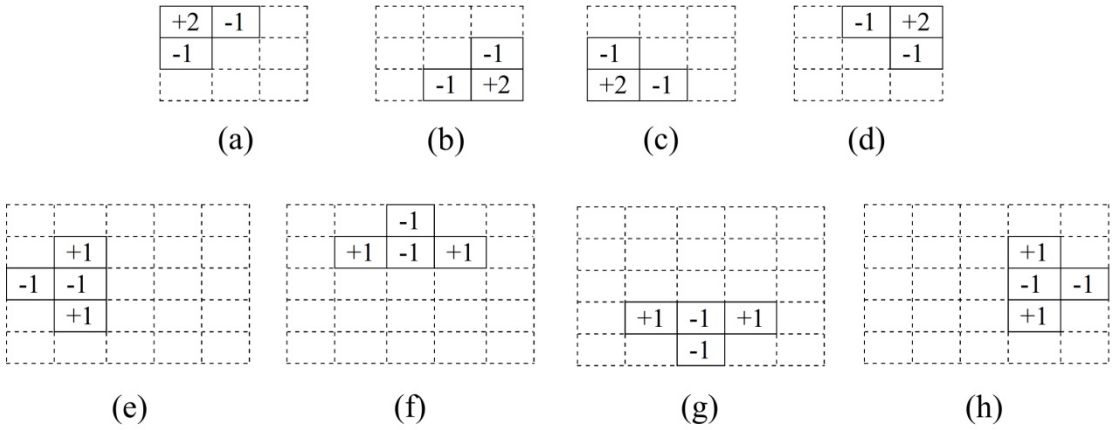

Figure 5.8 High-pass filters based on cliques.

5.6 Feature Construction and Experimentation

After the discussion of a variety of features made in Sections 5.2 to 5.5, one can observe that there are multiple ways to construct a feature set for steganalysis. An effective combination of features with a dimensionality of 19,593 is formulated based on the description in Sections 5.2 to 5.5. It is by no means to claim that this is the best possible combination of features in the proposed framework. Evaluated by FLD-based ensemble classifier with 12 independent runs, the proposed combination of features attains the average ensemble TP rate = 82.71%, TN rate = 82.50%, and AC = 82.60% at d_{red} = 2,800 and L = 65.

5.6.1 Feature Construction

The details of the proposed combination are summarized in Table 5.5.

Table 5.5 Details of the Proposed Feature Set

Features Described in Section	LBP Operators	Comments
5.3	Multi-resolution LBP: $LBP_{8,1}^{u2}$ +$LBP_{8,2}^{u2}$ +$LBP_{8,3}^{u2}$ (177-D features extracted from each residual image)	Successive prediction error PE_n (n = 1 to 5) (PEs) (885 features) and their variance images (VARpe) (885 features) Median-filter-based prediction error (MEDpe): all residual images were generated according to Table. 2 (1,593 features)
5.4		All the residual images were generated according to Table 5.3. (LMbased) (12,390 features)
5.5	The original LBP (LBP_8) (256-D features extracted from each residual image)	All the residual images are generated according to Figures 5.7 (MN7) (1,792 features) and 5.8 (CL8) (2,048 features).

5.6.2 Empirical Validations on PEs and VARpe Features

Simple scenarios are used to validate the concept of successive prediction error images, contrast measure, and multi-resolution approach. In this sub-section, FLD has been employed for 12 independent runs.

Table 5.6 Empirical Validations on PEs and VARpe

Residual	PE_1	PE_1-PE_2	PE_1-PE_3	PE_1-PE_4	PE_1-PE_5	$PEVAR_1$-$PEVAR_5$	
AC	61.29%	66.96%	68.49%	70.00%	70.78%	73.12%	76.55%
D	59	118	177	236	295	590	1,770
R	1	1	1	1	1	1	1, 2, 3

All the LBP operators used to construct features in Table 5.6 are based on uniformity mapping with $P = 8$ and different combination of R's. D stands for feature dimensionality. Note that the last column in Table 5.6 represents the multi-resolution setting of LBP operators ($LBP_{8,1}^{u2}$+ $LBP_{8,2}^{u2}$+ $LBP_{8,3}^{u2}$). In Table 5.6, PE_1-PE_5 and $PEVAR_1$-$PEVAR_5$ means that PE_1 to PE_5 and PE_1 to PE_5 together with their variance images are used as inputs to LBP operators, respectively. The statistics shown in Table 5.6 reveals the successive applications of the prediction error schemes, contrast measure, and multi-resolution approach of LBP have all contributed to enhance the detection accuracy.

5.6.3 Empirical Validations on Individual Type of Features

The individual performance of each type of features is evaluated in this sub-section. All types but LMbased is evaluated by using FLD while LMbased by using FLD-based ensemble classifiers. The average accuracies are computed over three independent runs of chosen classifier.

Table 5.7 FLD Accuracies of PEs, VARpe, MEDpe, MN7, and CL8

Feature Type	PEs	VARpe	MEDpe	MN7	CL8
AC	74.78%	68.05%	66.78%	71.08%	71.57%
Dimensionality	885	885	1,593	1,792	2,048

Evaluated by FLD-based ensemble classifiers with three independent runs, the accuracy (AC) of LMbased is 81.88% at d_{red} = 2,800 and L = 89.

5.6.4 Empirical Validations on Feature Elimination

In order to validate whether or not each type of features are essential to the final accuracy of the whole feature set, the performances of the whole feature sets as well as the whole feature sets with each individual type of features dropped out are evaluated and shown in Table 5.8. Note that each performance is evaluated over three independent runs of FLD-based ensemble classifier.

Table 5.8 Performance on Feature Elimination

Feature Set	Dimensionality	AC	d_{red}	L	Δ
Whole	19,593	83.27%	2,800	95	0.00%
Whole - PEs	18,708	82.82%	2,800	35	-0.45%
Whole - VARpe	18,708	82.75%	2,800	73	-0.52%
Whole - MEDpe	18,000	82.85%	2,800	67	-0.42%
Whole - LMbased	7,203	81.42%	1,800	49	-1.85%
Whole - MN7	17,801	83.02%	2,800	59	-0.25%
Whole – CL8	17,545	82.85%	2,800	29	-0.42%

The statistics in Table 5.8 reveals that each type of the proposed features is essential to the final accuracy. That is, the final accuracy decreases upon the absence of each type of features. The degree of contribution among all types of features can be ranked in descending order as follows: LMbased, VARpe, PEs, MEDpe (tied with CL8), and MN7. Note that it is very difficult to make a significant progress when more than

80% of detection accuracy has been attained. Therefore, only a fraction of percentage gained in the detection accuracy by some set of features matters in detection HUGO with high fidelity.

5.7 Discussion and Conclusions

The fact that a few powerful textural features, e.g., the transition probability matrix resulted from Markov process model, the co-occurrence matrix, and gray level run-length [21] have been successfully utilized in modern steganalysis, is the motivation to study more on textural features.

Two observations to be described lead to the increase in motivation. One observation made five to seven years ago [74, 75] is that steganalysis becomes more difficult if the data are hidden inside the texture images. Another recent observation is that steganalysis encounters challenge if the data are hidden in the area where statistical modeling is difficult. These are the key reasons to learn from the rich literatures in the field of texture classification to further enhance the steganalysis capability.

In this doctoral research, the first-stage study and investigation along this line are reported, specifically, on the potentials of local binary patterns (LBP) [79, 80] in steganalysis. These well-known techniques are inspired by Haralick et al.'s well-known co-occurrence matrix. In this LBP technique each pixel is compared with its neighbor pixels. Depending on if the neighbor pixel's gray value is larger or smaller than the central pixel's gray value or not, a binary zero or one is assigned to this neighboring pixel. This process is conducted for each pixel in a given image (or a region of interest). All of bins of the resultant histogram are used as LBP features. Furthermore, a multi-

resolution structure can be constructed by using multi-size neighbor, e.g., 3×3, 5×5 and 7×7. In addition to the LBP, the variance generated from the local neighborhood can also be used to characterize the contrast of the local region of, say, 3×3. This work has verified that the LBP, variance and multi-resolution do work well in steganalysis. In the test presented in Table 5.1, it is shown that even with 59 features the LBP outperform the co-occurrence matrix with 81 features in the BOSSbase 0.92 [72, 73] (In all experiments, the stego images are generated by HUGO at 0.4 bpp with default parameters). As to use the 256 bins or the 59 bins (the latter results from the so-called uniform mapping) in steganalysis, it depends. The experimental works have demonstrated that the selection of 256 bins often perform better than 59 bins if the feature dimensionality is low. As the dimensionality increases, this may change. Hence in this work both 256 bins and 59 bins are used for different kinds of features and scenarios.

Prior to further summarizing this work, it is essential to bring one point to readers' attention. That is, Avcibas et al. [83] proposed a steganalysis scheme which employs 18 binary similarity measures on the seventh and eighth bit planes in an image as distinguishing features. Instead of comparing, say, in a 3×3 neighborhood, the eight neighboring pixel values with the central pixel value to produce an eight-bit binary number so as to establish a histogram of 256 bins for classification, the authors [83] simply use the two least significant bit-planes in a given image without binarization. Furthermore, the bit corresponding to the central pixel position is included to formulate a nine-bit string, thus resulting in a histogram of 512, instead of 256, bins. One more difference is that the 59 and/or 256 features as suggested in the LBP technologies [79, 80] are used in this work, while the authors [83] compute four binary similarity measures on

the resulting 512-bin histograms [83] as features for steganalysis. Consequently, one should not consider the scheme in [83] as an application of the LBP technology.

Markov neighborhoods with Markov parameters utilized in Markov Random Field as shown in Figures 3.53, 3.60 and some of their cliques shown in Figure 3.68 in [78] have been studied in this chapter. Many of them with some addition as shown in Figures 5.7 and 5.8 have been used in the proposed steganalysis scheme.

Among Laws' masks as shown in Figures 4.126, 4.127 and 4.128 in [78], all the masks that are considered low-pass filters are excluded. Instead only the masks, which are considered high- pass filters, are used. To construct the even-number masks to boost steganalyzing capability, the well-known [-1 1] mask is used as a two-tap mask to convolute with the *S3* (one kind of Laws' mask), i.e., [-1 2 -1] to form by the four-tap mask. The six-tap mask is formulated in the similar fashion. The experimental works have verified the contribution made by these masks.

Intensive studies as well as experiments leads to the achieved average detection accurate rate of 82.60% in the BOSSbase 0.92 [72, 73]. This has indicated that the proposal to utilize techniques developed in the field of texture classification for steganalysis is effective.

This piece of work, however, may not present the optimal feature set construction for steganalysis on HUGO. It may be possible to construct a more effective set of feature along the line of the proposed framework; additionally, the framework itself may be enhanced either by adding some new elements on or dropping some existing ones out. Last but not least, some feature selection techniques should be investigated to reduce the number of features which may, as a result, boost the performance of the scheme.

CHAPTER 6

IMAGE RECAPTURE DETECTION

This chapter presents an effective method for image recapture detection. The proposed scheme is developed based in the knowledge obtained in the study of textural features heavily discussed in Chapter 5. The effectiveness of the proposed method is then evaluated over a public dataset for smartphone image recapture detection evaluation. The outcome of the evaluation reveals that the proposed methods perform better than the related prior art.

6.1 Introduction

Nowadays, digital cameras are ubiquitous as they have been integrated in smartphones, tablet computers, laptop computers, or even ATM machines. One of side benefits of the ubiquity of digital camera is the potential to strengthen the security system of various systems especially in terms of face authentication. Some companies, for instance [8, 9], have already realized the potential of face authentication system and established face recognition based access control system.

Along the afore-mentioned side benefits comes the possible threat of faked identity attack in which recaptured face photo could be used to fool around the security system. As a result, image recapture detection (IRD) technology, which aims at differentiating real scenes from the recaptured images, has been called for to protect such systems.

Ng et al. [42] proposed to use geometry features to differentiate photographic images from computer graphics. Part of computer graphics used in the experiment in [42] was recaptured from an LCD screen; therefore, the effectiveness of geometry features also implies their discriminability between photographic and recaptured images. Yu et al. [84] devised a feature from the underlying statistical difference between photographic and recaptured images from the specularity component. Gao et al. [85] proposed an IRD scheme utilizing a general physical model developed based on several image properties such as the contextual background information, the spatial distribution of specularity, the image gradient, the color information as well as contrast, and the blurriness measure. Gao et al. [86] later released I^2R Open Dataset for Smart-phone Recaptured Images [87], a smartphone image dataset for IRD, and re-evaluated the image features in [42, 85, 88, 89] over the newly constructed dataset in order to draw a fair comparison among different kinds of image features.

Cao and Kot [90] proposed three types of features to distinguish real-scene images from images recaptured from LCD screens over the closed dataset specifically constructed. The first type of features is based on multi-resolution local binary pattern (LBP) [80]. The details of LBP operators are briefly described in Chapter 5 and the generic notation of LBP descriptor is $LBP_{P,R}^{mapping}$ where P and R is the number of neighbor pixels and the radius of the circular symmetric neighborhood, respectively. The LBP-based features in [90] is extracted directly from the input image and derived entirely using rotation invariant and uniformity mapping or "riu2" and consist of the following combination of four LBP operators: $LBP_{8,1}^{riu2}$, $LBP_{16,2}^{riu2}$, $LBP_{24,3}^{riu2}$, and $LBP_{24,4}^{riu2}$. Such four LBP operators result in 80-D features, denoted as MRLBP-80, which yield the

detection accuracy of 99.1%. In the second type of features, per color channel in RGB space, the mean and standard deviation of the absolute Haar wavelet coefficients from high-frequency sub-bands (*HL*, *LH*, *HH*) of the first three levels are computed as distinguishing features. All three channels are then used to form 54 features which yield the detection accuracy of 98.9%. The third type of features is designed based on color anomalies introduced in recaptured images and consists of 21 features which yield the detection accuracy of 82.60%. Such features are a collection of various image properties including three average pixel values, three pair correlations, three neighbor distribution centers of mass, three pairs energy ratios [91] from the RGB color space and nine color moments from HSV color space [92].

6.2 Image Features

In this chapter, an effective scheme based textural features is proposed for IRD based on a combined usage of Laws' mask and single-resolution LBP operator which have intensively discussed in Chapter 5.

6.2.1 Residual Image

It is well known that effective image features should be barely influenced by image content, and a common effective means of eliminating image content is to apply a high-pass filter onto a given image.

In this research, a 3×3 high-pass filter is proposed to use to remove image content from the input image before feature extraction. The 3×3 high-pass filter is denoted by W_{S3} formulated by Spot 3 (*S3*), a three-tap Laws' mask [78], is proposed to use. The formulation of W_{S3} is given in Equation 6.1. $R(x,y)$, the resultant residual image

obtained by filtering the input image luminance $I(x,y)$ with W_{S3}, can be expressed in Equation 6.2. Note that $R(x,y)$ can also be found by applying $S3$ in the horizontal direction onto $I(x,y)$ and then in the vertical direction in a cascaded manner.

$$W_{S3} = S3^T \times S3 = \begin{bmatrix} -1 & 2 & -1 \end{bmatrix}^T \times \begin{bmatrix} -1 & 2 & -1 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \tag{6.1}$$

$$R(x,y) = W_{S3} * I(x,y) \tag{6.2}$$

where * denotes spatial filtering operator.

6.2.2 Local Binary Pattern Features

High-pass filtering an image luminance results in a residual image which is a representation of a statistical spatial noise pattern. Such a pattern depends little to image content and is vital information to IRD. To find a compact representation of such pattern for IRD, local binary pattern (LBP) is employed to model the spatial noise statistics and the normalized LBP histogram is used as distinguishing features.

The proposed features, relying on uniformity mapping or "u2" which is found to be generally more powerful in this application than "riu2," are derived by applying $LBP_{8,1}^{u2}$ onto a residual image $R(x,y)$. As discussed in Chapter 5, the output of $LBP_{8,1}^{u2}$ is 59-bin histogram; mathematically, the sum of the frequency of occurrence of all the bins depends on image sizes. Since the sizes of real-scene and recaptured images in the dataset [87] to be used for IRD evaluation are different, the resultant histogram is then

normalized such that the sum of the frequency of occurrence of all the bins equals to one in order to suppress the influence of different image sizes to classification performance. All the elements of normalized LBP histograms are finally used to form 59-D features for IRD.

In this research work, the proposed features, denoted as LMLBP-59 and depicted in Figure 6.1, as well as MRLBP-80, the relevant features in [90], are evaluated over the publicly available dataset [87]. Note that all the LBP histograms used to form MRLBP-80 are also normalized in the fashion as in LMLBP-59.

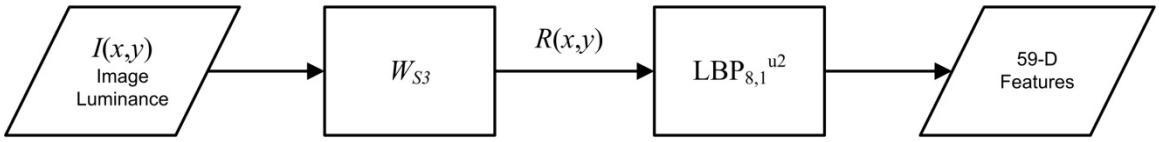

Figure 6.1 Block diagram of LMLBP-59 feature extraction process.

6.3 Image Datasets

There is currently only one public dataset [87] indirectly related to image recapture detection. In [87], real-scene images are taken using either a smartphone camera at VGA resolution or a high-end DSLR (Digital Single-Lens Reflex) camera. The cameras built-in the following three smartphones are used: Acer M900, Nokia N95, and HP iPAQ hw 6960. There are also three different DSLR cameras used which are Nikon D90, Canon EOS 450D, and Olympus E-520.

Real-scene images captured by using a DSLR camera were displayed on an LCD display or on a printed medium and then recaptured using the same camera having used in capturing the real-scene images. That is, recaptured images are the fruit of the

reproduction process recently described. There are three different kinds of LCD displays involved in the creation of the dataset: Dell 2007FP LCD screen (1600×1200 pixels), Acer M900 smartphone screen, and iPhone 3GS smartphone screen. For the printing reproduction process, some real-scene images are printed either on A4-size plain paper, 4R glossy, or matte photo by using any of the following printers: HP CP3505dn laser printer and Xerox Phaser 8400 ink printer. In the dataset creation, some post-processing operations are then applied to resultant recaptured images from the reproduction process. The dataset comprises four sub-datasets: 1) Real-Scene Dataset (for Recaptured Dataset A); 2) Recaptured Dataset A; 3) Recaptured Dataset B; 4) Recaptured Dataset C. The brief dataset details are presented in Section 6.3.1.

6.3.1 Real-Scene Image Dataset

Since the creators of the dataset [87] focus on the application of IRD in smartphones, the Real-Scene Dataset is then generated entirely using smart-phone cameras. There are totally 1,094 real-scene images.

Figure 6.2 Examples of images in Real-Scene Dataset [87].

6.3.2 Recaptured Dataset A

Each image contains recaptured image roughly at the center portion and real environment background along its borders. There are 1,137 images in total.

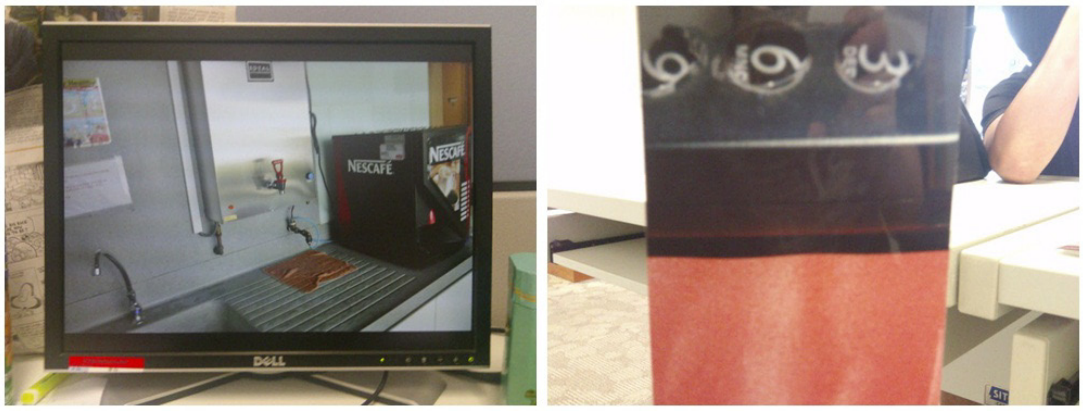

Figure 6.3 Examples of images in Recaptured Dataset A. (left) Recaptured image from an LCD Screen; (right) recaptured image from a printed medium [87].

6.3.3 Recaptured Dataset B

The images in Recaptured Dataset A are cropped by using Matlab to remove the real environment background. This post-processing makes the size of a recaptured image dependent on the recaptured image portion present in the corresponding image in Recaptured Dataset A. There are 1,765 recaptured images in total.

Figure 6.4 Examples of images in Recaptured Dataset B (no real-scene environment) [87].

6.3.4 Recaptured Dataset C

In total, 587 recaptured images are cropped to remove the real environment scene and then homography transformed.

Figure 6.5 Examples of images in Recaptured Dataset C (cropped and homography transformed) [87].

6.4 Experimentation

The initial work on IRD in [86] reports the performance of four statistical images features [42, 85, 88, 89] on Recaptured Datasets A, B, and C. The average detection accuracy over the whole dataset is 78.49% for [89], 85.26% for [42] , 84.28% for [85], and 74.84% for [88].

In [86], real-scene images are post-processed for the evaluation in Recaptured Datasets B (cropping to match the image sizes) and C (cropping and homography transforming); however, the post-processed real-scene images are not publicly available in [87] and can hardly be generated in the exact same way. This is not only because of the post-processing but also of the unavailability of real-scene images taken by using DSLR cameras. Moreover, the post-processing in [86] are likely to introduced some non-

intrinsic artifacts to real-scene images which is viewed, in this research work, as an alteration in the definition of real-scene images. Recaptured Dataset A is, however, excluded from the evaluation in this research work as the recaptured images in the dataset are mixed with the real-scene images which, in some aspect, may be considered irrelevant to the very nature of IRD.

In this research work, original real-scene images in Real-Scene Dataset are then used to for the evaluation of LMLBP-59, the proposed scheme, in comparison with MRLBP-80 [90]. Note that the usage of normalized LBP histogram in the generation of LMLBP-59 and MRLBP-80 can substitute the post-processing on real-scene images used to evaluate Recaptured Dataset B in [86] as it eliminates the influence of image size on the classification.

The way of training and testing image formulation in this work can still be deemed as the exact representation of IRD problem although in a different scenario from that in [86]. Therefore, the obtained results are by no means to directly compare with those reported in [86].

Support vector machines (SVM) [36] are employed with RBF kernel for 10 independent run with random data partition (four-fifths for training and one-fifth for testing). Optimal kernel parameters are obtained by coarse grid searching with five-fold cross validation. The average detection accuracies of MRLBP-80 and LMLBP-59 are shown in Tables 6.1 and 6.2. Note that recaptured and real-scene images are considered positive and negative instances, respectively.

Table 6.1 Detection Rates of Real-Scene Dataset and Recaptured Dataset B

Feature Type	MRLBP-80	LMLBP-59
TP Rate	94.39%	96.01%
TN Rate	88.40%	95.98%
AC	92.09%	96.00%
Dimensionality	80	59

Table 6.2 Detection Rates of Real-Scene Dataset and Recaptured Dataset C

Feature Type	MRLBP-80	LMLBP-59
TP Rate	89.49%	92.99%
TN Rate	96.58%	98.31%
AC	94.11%	96.46%
Dimensionality	80	59

6.5 Discussion and Conclusions

In this chapter, a compact and effective set feature is proposed for image recapture detection (IRD) based on the textural feature framework successfully applied to solve advanced steganalysis problem in Chapter 5. Of 59-D, the proposed features are generated from a given image luminance by the application of Laws' mask to remove the influence of image content and local binary pattern (LBP) operator to model image statistics.

The LBP operator used in the generation of the proposed 59-D features is a single-resolution approach based on uniformity mapping (u2) and can be described in the LBP notation as $LBP_{8,1}^{u2}$. The proposed features, consisting of 59 features and denoted as LMLBP-59, are directly compared with MRLBP-80, 80-D LBP features proposed in [90] formulated in a multi-resolution approach using four different kinds of LBP operators all with rotation invariant and uniformity mapping (riu2): $LBP_{8,1}^{riu2}$, $LBP_{16,2}^{riu2}$, $LBP_{24,3}^{riu2}$, and $LBP_{24,4}^{riu2}$.

The outcome of the evaluation reveals that LMLBP-59, although of less dimensionality, performs better than MRLBP-80 remarkably, which implies the combined usage of the proposed high-pass filtering technique and LBP with u2 mapping is more powerful than the only usage of LBP with riu2 mapping under this circumstance. The phenomena that LBP features generated from u2 mapping on residual images is stronger than those from riu2 have also been observed in other scenarios such as in steganalysis.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

This dissertation presents two image statistical frameworks, one in the frequency domain, the other in the spatial domain. The construction of an image statistical feature set is viewed as an open process. That is, some elements in the framework of interest can be mixed together to form an effective set of image statistical features for a specific application. Each of the two frameworks is found effective in two digital image forensic applications. In total, four digital image forensic problems are investigated in this dissertation. The pitfalls of related JPEG-domain (frequency-domain) techniques are intensively investigated by the anti-forensic tools proposed in this doctoral research work which reveals some vital implicit information on the practicality of digital image tampering detection schemes in the JPEG domain. The knowledge gained by anti-forensic research in this dissertation is potentially fruitful to the design of the next generation of digital image forensic schemes.

Markovian rake transform (MRT), the frequency-domain image statistical framework, employs block discrete cosine transform in a multi-resolution way as well as Markov modeling. MRT framework, a sub-framework within the general natural image framework proposed in [29], is effective in both digital image tampering detection and classification of computer graphics and photographic images. The attempts to test real-life tampered images collected from the Internet are made to study the reliability of the digital image tampering detection schemes discussed in this doctoral research as well as its potentiality in being rendered a practical application. The outcomes of such testing

suggest a low degree of reliability of the proposed schemes, thus limiting the practicality of the schemes. One of the possible explanations could be because of the utilization of first order difference which appears to make the statistical features much influenced by image contents. Another possible explanation is to be discussed below.

The endurance of irregularities in the JPEG domain is studied in a relatively simple scenario. That is, a few proposed attacks, based on non-malicious common image processing operations, are individually applied to doubly JPEG compressed images in order to learn the endurance of double compression (DQ) artifacts characterized by a peak-and-valley pattern in a JPEG mode histogram. The consequence of such attacks, in general, is the suppression of DQ artifacts. It is well known that irregularities in the frequency domain left by tampering operation on compressed images are vital information for detection tampered images and that non-malicious common image operations such as image resizing are frequently applied to digital images including tampered ones before distribution; as a result, the JPEG-domain artifacts caused by image tampering can hardly be relied upon which limits the practicality of digital image tampering detection schemes in the JPEG domain.

The spatial-domain image statistical framework is proposed based on several textural operators. The key mechanism of the framework is the application of local binary pattern (LBP) [79, 80] operators to model local image statistics. LBP may be viewed as a simplified version of the eighth-order co-occurrence matrix which makes it generally more powerful than the co-occurrence matrix popularly used in forensic research area. Effective features should be less influenced by image contents, so LBP [79, 80] operators are applied to a number of residual images, many of which are generated by popular tools

in texture analysis such as Laws' mask, Markov Random Field neighborhoods, and cliques. Most of the resultant residual images are considered higher-order, having little influence of image contents. The textural feature framework is found effective on steganalysis on HUGO (Highly Undetectable Steganography) [7] which embedded hidden messages locally into image regions difficult to model statistics and image recapture detection (IRD). The localization capability of LBP-based features is confirmed in the steganalysis problem while the effectiveness of such features is also present in the application like IRD in which the difference in the underlying image statistics between different classes is considered homogenous. It is therefore expected that the proposed textural feature framework can be effectively utilized for many other digital image forensic problems such as classification of computer graphics and photographic images, digital image tampering detection, and so on. However, a better way to construct a feature set as well as some feature selection techniques should be investigated; additionally, the framework itself may be enhanced by either adding some new elements on or dropping some existing ones out.

The ultimate goal of digital image forensic research is certainly to move the development of digital forensic schemes onto the practical level, in other words, to render real-life applications with high fidelity. To achieve such a goal, such schemes must be robust against various kinds of attacks, especially against non-malicious common image processing operations. Therefore, the study on the robustness of image statistical features should be emphasized along the line of the development of digital forensic schemes.

REFERENCES

[1] The Air Force Research Laboratory, available from: https://www.fbo.gov/index?print_preview=1&s=opportunity&mode=form&id=883ef2b019e735fc63407f3c4bfa4a1f&tab=core&tabmode=list (access in April 2012).

[2] The Guardian, available from: http://www.guardian.co.uk/world/2012/jan/09/russian-navalny-fake-photo-smear (access in April 2012).

[3] J. Sun, J. Jia, C.K. Tang, and H.Y. Shum, "Poisson matting," *ACM SIGGRAPH*, pp. 315-321, Los Angeles, California, 2004.

[4] Y. Li, J. Sun, C.K. Tang, and H.Y. Shum, "Lazy snapping," *ACM SIGGRAPH*, pp. 303-308 Los Angeles, California, 2004.

[5] Cyber-Rights & Cyber-Liberties (UK), available from: http://www.cyber-rights.org/reports/uscases.htm (access in April 2012).

[6] USA Today, available from: http://www.usatoday.com/life/cyber/tech/2001-02-05-binladen.htm (access in April 2012).

[7] T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," *12th International Conference on Information Hiding*, pp. 161-177, Calgary

[8] Korea identication inc, available from: http://www.korea-id.co.kr/eng/index.html (accessed in November 2011).

[9] XID technologies, available from: http://www.xidtech.com/ (accessed in November 2011).

[10] Z. Zhang, G. Qiu, Q. Sun, X. Lin, Z. Ni, and Y.Q. Shi, "A unified authentication framework for JPEG2000," *IEEE International Conference on Multimedia and Expo*, pp. 915-918, Taipei, Taiwan, June 2004.

[11] T.T. Ng, S.F. Chang, C.Y. Lin, and Q. Sun, "Passive-blind image forensics," *Multimedia Security Technologies for Digital Rights*, W. Zeng, H. Yu, and C.Y. Lin, Editors, pp. 383-412, Academic Press, Missouri, 2006.

[12] P. Sutthiwan, Y.Q. Shi, W. Su, and T.T. Ng, "Rake transform and edge statistics for image forgery detection," *IEEE International Conference on Multimedia and Expo*, pp. 1463-1468, Singapore, July 2010.

[13] P. Sutthiwan, Y.Q. Shi, H. Zhao, T.T. Ng, and W. Su, "Markovian rake transform for digital image tampering detection," *Transactions on Data Hiding and Multimedia Security VI*, pp. 1-17, 2011.

[14] T.T. Ng, S.F. Chang, and Q. Sun, "Blind detection of photomontage using higher order statistics," *IEEE Internation Symposium on Circuits and Systems*, vol. 5, pp. 688-691, Vancouver, Canada, May 2004.

[15] M.K. Johnson, and H. Farid, "Exposing digital forgeries by detecting inconsistencies in lighting," *7th ACM Workshop on Multimedia and Security*, pp. 1-10, New York, New York, August 2005.

[16] M.K. Johnson, and H. Farid, "Exposing digital forgeries in complex lighting environments," *IEEE Transaction on Information Forensics and Security*, vol. 2, no. 3, pp. 450-461, 2007.

[17] Y.F. Hsu, and S.F. Chang, "Detecting image splicing using geometry invariants and camera characteristics consistency," *IEEE International Conference on Multimedia and Expo*, pp. 549-552, Toronto, Ontario, Canada, July 2006.

[18] D. Fu, Y.Q. Shi, and W. Su, "Detection of image splicing based on Hilbert-Huang transform and moments of characteristic functions with wavelet decomposition," *5th International Workshop on Digital Watermarking*, pp. 177-187, Jeju Island, Korea, November 2006.

[19] W. Chen, Y.Q. Shi, and W. Su, "Image splicing detection using 2-D phase congruency and statistical moments of characteristic function, *Proceeding of SPIE*, vol. 6505, San Jose, California, February 2007.

[20] P. Sutthiwan, Y.Q. Shi, J. Dong, T. Tan, and T.T. Ng, "New developments in color image tampering detection," *IEEE International Symposium on Circuits and Systems*, pp. 3064-3067, Paris, France, May 30-June 2, 2010.

[21] J. Dong, W. Wang, T. Tan, and Y.Q. Shi, "Run-length and edge statistics based approach for image splicing detection," *8th International Workshop on Digital Watermarking*, pp. 76-87, Guildford, United Kingdom, August 2009.

[22] H. Farid, "Exposing digital forgeries from JPEG ghosts," *IEEE Transaction on Information Forensics and Security*, vol. 4, no. 1, pp. 154-160, 2009.

[23] Z. Qu, G. Qiu, and J. Huang, "Detect digital image splicing with visual cues," *11th International Workshop on Information Hiding*, pp. 247-261, Darmstadt, Germany, June 2009.

[24] A.E. Dirik, and N. Memon, "Image tamper detection based on demosaicing artifacts," *16th IEEE International Conference on Image Processing*, pp. 1497-1500, Cairo, Eqypt, November 2009.

[25] W. Wei, J. Dong, and T. Tan, "Effective image splicing detection based on image chroma, *16th IEEE International Conference on Image Processing*, pp. 1257-1260, Cairo, Eqypt, November 2009.

[26] W. Wei, J. Dong, and T. Tan, "Image tampering detection based on stationary distribution of Markov chain," *17th IEEE International Conference on Image Processing*, pp. 2101-2104, Hong Kong, September 2010.

[27] DVMM Laboratory of Columbia University, Columbia image splicing detection evaluation dataset, available from: http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSpliced-DataSet/AuthSplicedDataSet.htm (accessed in September 2007).

[28] CASIA tampered image detection evaluation database, available from: http://forensics.idealtest.org (accessed in January 2010).

[29] Y.Q. Shi, C. Chen, and W. Chen, "A natural image model approach to splicing detection," *9th ACM Multimedia and Security Workshop*, pp. 51-62, Dallas, Texas.

[30] Y.Q. Shi, C. Chen, G. Xuan, and W. Su, "Steganalysis versus splicing detection, " *6th International Workshop on Digital Watermarking*, pp. 158-172, Guangzhou, China, November 2007.

[31] C. Chen, "Detecting hidden information in digital images, " *Ph.D. Dissertation*, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, New Jersey, 2008.

[32] Y.Q. Shi, C. Chen, and W. Chen, "A Markov process based approach to effective attacking JPEG steganography," *8th International Conference on Information Hiding*, pp. 249-264, Alexandria, Virginia.

[33] A. Leon-Garcia, "Probability and random processes for electrical engineering," 2nd Edition, Addison-Wesley, 1993.

[34] Impulse Adventure, available from: http://www.impulseadventure.com/photo/jpeg-quality.html (accessed in September 2010).

[35] G. Schaefer, and M. Stich, "UCID: an uncompressed color image database," *Proceeding of SPIE*, vol. 5307, pp. 472-480, San Jose, California, 2003.

[36] C.C. Chang, and C.J. Lin, "LIBSVM : a library for support vector machines, *ACM Transactions on Intelligent Systems and Technology*, 2011, available from: http://www.csie.ntu.edu.tw/~cjlin/libsvm (accessed in December 2011).

[37] Photo Tampering throughout History, available from: http://www.fourandsix.com/photo-tampering-history/?currentPage=14 (accessed in December 2010).

[38] P. Sutthiwan, X. Cai, Y.Q. Shi, and H. Zhang, "Computer graphics classification based on Markov process model and boosting feature selection technique," *16th IEEE International Conference on Image Processing,* pp. 2913-2916, Cairo, Eqypt, November 2009.

[39] P. Sutthiwan, J. Ye, and Y.Q. Shi, "An enhanced statistical approach to identifying photorealistic images," *8th International Workshop on Digital Watermarking*, pp. 323-335, Guildford, United Kingdom, August 2009.

[40] T.I. Ianeva, A.P. de Vries, and H. Rohrig, "Detecting cartoons: a case study in automatic video-genre classification," *IEEE International Conference in Multimedia and Expo*, pp. 449-452, Baltimore, Maryland, July 2003.

[41] S. Lyu, and H. Farid, "How realistic is photorealistic?," IEEE Transactions on Signal Processing, vol. 53, no. 2, pp. 845-850, 2005.

[42] T.T. Ng, S.F. Chang, J. Hsu, L. Xie, and M.P. Tsui, "Physics-motivated features for distinguishing photographic images and computer graphics," *13th Annual ACM*

International Conference on Multimedia, pp. 239-248, Singapore, November 2005.

[43] W. Chen, Y.Q. Shi, and G. Xuan, "Identifying computer graphics using HSV color model and statistical moments of characteristic functions," *IEEE International Conference on Multimedia and Expo*, pp. 1123-1126, Beijing, China, July 2007.

[44] W. Chen, "Detection of digital image and video forgeries," *Ph.D. Dissertation*, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, New Jersey, 2008.

[45] F. Friedma, and T. Hastie, "Additive logistic regression: a statistical view of boosting," *An Official Journal of the Institute of Mathematical Statistics*, vol. 28, no. 2, pp. 337-407, 2000.

[46] K. Tieu, and P. Viola, "Boosting image retrieval," *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 228-235, Barcelona, Spain, September 2000.

[47] Creative 3D, available from: http://www.creative-3d.net (accessed in September 2008).

[48] Ultimate 3D links, available from: http://www.3dlinks.com/ (accessed in September 2008).

[49] DVMM Laboratory of Columbia University, Columbia photographic images and photorealistic computer graphics dataset, available from: http://www.ee.columbia.edu/ln/dvmm/downloads/PIM_PRCG_dataset/ (accessed in September 2008).

[50] P. Sutthiwan, and Y.Q. Shi, "Anti-forensics of double JPEG compression detection," *10th International Workshop on Digital-Forensics and Watermarking*, pp. 411-424, Atlantic City, New Jersey, October 2011.

[51] M.C. Stamm, S.K. Tjoa, W.S. Lin, and K.J.R. Liu, "Anti-forensics of JPEG compression," *IEEE International Conference on Acoustics Speech and Signal Processing*, pp. 1694-1697, Dallas, Texas, March 2010.

[52] M.C. Stamm, S.K. Tjoa, W.S. Lin, and K.J.R. Liu, "Undetectable image tampering through JPEG compression anti-forensics," *17th IEEE International Conference on Image Processing*, pp. 2109-2112, Hong Kong, September 2010.

[53] M.C. Stamm, and K.J.R. Liu, "Anti-forensics of digital image compression," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1050-1065, 2011.

[54] A.C. Popescu, "Statistical tools for digital image forensics," *Ph.D. Dissertation*, Department of Computer Science, Dartmouth College, Hanover, New Hampshire, 2005.

[55] F.J., Huang, and Y.Q. Shi, "Detecting double JPEG compression with the same quantization matrix," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 848-856, 2010.

[56] C. Chen, Y.Q. Shi, and W. Su, "A machine learning based scheme for double JPEG compression detection," 19th IEEE International Conference on Pattern Recognition, pp. 1-4, Tampa, Florida, December 2008.

[57] B. Li, Y.Q. Shi, and J. Huang, "Detecting doubly compressed JPEG images by using mode based first digit features, *10th IEEE International Workshop on Multimedia Signal Processing*, pp. 730-735, Cairns, Queemsland, Australia, October 2008.

[58] J. Fridrich, M. Goljan, and D. Hogea, "Steganalysis of JPEG images: Breaking the F5 Algorithm," *5th International Workshop on Information Hiding*, pp. 310-323, Noordwijkerhout, The Netherlands, October 2002.

[59] J. Fridrich, "Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes," *6th International Workshop on Information Hiding*, pp. 67-81, Toronto, Canada, May 2004.

[60] R.M. Haralick, K. Shanmugam, and I.H. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, no. 6, pp. 610-621, 1973.

[61] Y.Q. Shi, P. Sutthiwan, and L. Chen, "Textural features for steganalysis," *14th International Conference on Information Hiding*, Berkeley, California, May 2012 (accepted).

[62] I. Avcibas, N. Memon, and B. Sankur, "Steganalysis using image quality metrics," *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 221-229, 2003.

[63] S. Lyu, and H. Farid, Detecting hidden messages using higher-order statistics and support vector machines, *5th International Workshop on Information Hiding*, pp. 340-354, Noordwijkerhout, The Netherlands, October 2002.

[64] J.J. Harmsen, "Steganalysis of additive noise modelable information hiding, *M.S. Thesis*, Electrical, Computer and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, New York, 2003.

[65] G. Xuan, Y.Q. Shi, J. Gao, D. Zou, C. Yang, Z. Zhang, P. Chai, C. Chen, and W. Chen, "Steganalysis based on multiple features formed by statistical moments of wavelet characteristic functions," *7th International Conference on Information Hiding*, pp. 262-277, Barcelona, Spain, June 2005.

[66] K. Sullivan, U. Madhow, S. Chandrasekaran, B.S. Manjunath, "Steganalysis of spread spectrum data hiding exploiting cover memory," *Proceeding of SPIE* , vol. 5681, pp. 38-46, San Jose, California, January 2005.

[67] D. Zou, Y.Q. Shi, W. Su, and G. Xuan, "Steganalysis based on Markov Model of Thresholded Prediction-Error Image," *IEEE International Conference on Multimedia and Expo*, pp. 1365-1368, Toronto, Ontario, Canada, July 2006.

[68] T. Pevný, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *11th ACM Workshop on Multimedia and Security*, pp. 75-84, Princeton, New Jersey, September 2009.

[69] J. Fridrich, J. Kodovský, V. Holub, and M. Goljan, "Steganalysis of content-adaptive steganography in spatial domain," *13th International Conference on Information Hiding*, pp. 102-117, Prague, Czech Republic, May 2011.

[70] J. Fridrich, J. Kodovský, V. Holub, and M. Goljan, "Breaking HUGO: the process discovery," *13th International Conference on Information Hiding*, pp. 85-101, Prague, Czech Republic, May 2011.

[71] G. Gul and F. Kurugollu, A new methodology in steganalysis: breaking highly undetectable steganograpy (HUGO)," *13th International Conference on Information Hiding*, pp. 71-84, Prague, Czech Republic, May 2011.

[72] P. Bas, T. Filler, and T. Pevný, "Break our steganographic system": the ins and outs of organizing BOSS," *13th International Conference on Information Hiding*, pp. 59-70, Prague, Czech Republic, May 2011.

[73] T. Filler, T. Pevný, and P. Bas, *BOSS. 2010*, available from: http://www.agents.cz/boss/BOSSFinal/ (accessed in November 2011).

[74] R. Bohme, "Assessment of steganalytic methods using multiple regression models," *7th International Conference on Information Hiding*, pp. 278-295, Barcelona, Spain, June 2005.

[75] C. Chunhua, Y.Q. Shi, and G. Xuan, "Steganalyzing texture images," *IEEE International Conference on Image Processing*, vol. 2, pp. 153-156, San Antonio, Texas, September 2007.

[76] J.S. Weszka, C.R. Dyer, and A. Rosenfeld, "A Comparative Study of Texture Measures for Terrain Classification," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 6, no. 4, pp. 269-285, 1976.

[77] P.P. Ohanian, and R.C. Dubes, "Performance evaluation for four classes of textural features," *Pattern Recognition*, vol. 25, no. 8, pp. 819-833,1992.

[78] M. Petrou and P.G. Sevilla, "Image processing dealing with texture," John Wiley & Sons Inc, 2006.

[79] T. Ojala, M. Pietikainen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51-59, 1996.

[80] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987, 2002.

[81] M.J. Weinberger, G. Seroussi, and G. Sapiro, "LOCO-I: a low complexity, context-based, lossless image compression algorithm," *IEEE Data Compression Conference*, pp. 140-149, Snowbird, Utah, March 31 – April 3, 1996.

[82] Y.Q. Shi, G. Xuan, D. Zou, J. Gao, C. Yang, Z. Zhang, P. Chai, W. Chen, and C. Chen, "Image steganalysis based on moments of characteristic functions using wavelet decomposition, prediction-error image, and neural network, " *IEEE International Conference on Multimedia and Expo*, pp. 4-8, Amsterdam, The Netherlands, July 2005.

[83] I. Avcibas, M. Kharrazi, N. Memon, and B. Sankur, "Image steganalysis with binary similarity measures" EURASIP Journal on Applied Signal Processing, vol. 2005, pp. 2749-2757, 2005.

[84] H. Yu, T.T. Ng, and Q. Sun, "Recaptured photo detection using specularity distribution," *15th IEEE International Conference on Image Processing*, pp. 3140-3143, San Diego, California, October 2008.

[85] X. Gao, T.T. Ng, B. Qiu, and S.F. Chang, "Single-view recaptured image detection based on physics-based features," *IEEE International Conference on Multimedia and Expo*, pp. 1469-1474, Singapore, July 2010.

[86] X. Gao, B. Qiu, J.J. Shen, T.T. Ng, and Y.Q. Shi, "A smart phone image database for single image recapture detection," *9th International Workshop on Digital Watermarking*, pp. 90-104, Seoul, Korea, October 2010.

[87] I^2R Open Dataset for Smart-phone Recaptured Images, available from: http://www1.i2r.a-star.edu.sg/~ttng/download/Recaptured-images.zip (accessed in November 2011).

[88] C. Chen and Y.Q. Shi, "JPEG image steganalysis utilizing both intrablock and interblock correlations," *IEEE International Symposium on Circuits and Systems*, pp. 3029-3032, Seattle, Washington, May 2008.

[89] H. Farid and S. Lyu, "Higher-order wavelet statistics and their application to digital forensics," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 94-101, Madison, Wisconsin, June 2003.

[90] H. Cao and A.C. Kot, "Identification of recaptured photographs on LCD screens," *IEEE International Conference on Acoustics Speech and Signal Processing*, pp. 1790-1793, March 2010.

[91] K.L. Mehdi, H.T. Sencar, and N. Memon, "Blind source camera identification," IEEE International Conference on Image Processing, vol. 1, pp. 709-712, Singapore, October 2004.

[92] Y. Chen, Z. Li, M. Li, W.Y. Ma, "Automatic classification of photographs and graphics," *IEEE International Conference on Multimedia and Expo*, pp. 973-976, Toronto, Ontario, Canada, July 2006.