Integrity Verification of Digital Images through Fragile Watermarking and Image Forensics

A thesis submitted in accordance with the requirements of
the University of Liverpool for the degree of Doctor in Philosophy

by

Sergio Bravo-Solorio

September 2010

Declaration

The work in this thesis is based on research carried out at the University of Liverpool. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Sergio Bravo-Solorio
Liverpool, United Kingdom.

Abstract

Detecting manipulations in digital images by bare-eye examinations has become increasingly difficult, due to the availability of open-source and commercial image editing systems equipped with sophisticated processing tools. This has raised serious concerns about the use of digital images in many application domains, especially in fields where the visual content could be misinterpreted as a result of malicious, or even inadvertent, distortions.

This thesis focuses on two different approaches aimed at detecting manipulations in digital images, namely fragile watermarking and image forensics. Fragile watermarking describes methods that introduce imperceptible changes in digital images in order to embed information, which could be retrieved, at some point, to verify the integrity the protected (watermarked) images. Image forensics, on the other hand, describes techniques to detect possible evidence of tampering, without relying on any authentication information generated *a priori*.

Fragile watermarking methods can be designed not only to determine whether the watermarked image has been distorted at all, but also to localise the tampered regions and restore the original content. This work presents a secure fragile watermarking method with improved localisation capabilities. The scheme employs a different slant on verification, whereby the integrity of each non-overlapping block of pixels is not verified independently of the rest of the pixel blocks. Instead, the authentication information retrieved from all the blocks is examined to determine whether the image contains the expected watermark and localise altered regions.

Additionally, a fragile watermarking with improved tampering localisation and restoration capabilities is presented. This is achieved by combining a secure mechanism resilient to cropping, which localises altered blocks of pixels, with an additional mechanism that enhances the localisation and estimates the original value of watermarked pixels in altered regions. The proposed scheme provides flexibility to control the trade-off between the embedding distortion and the localisation/restoration performance of the system. Moreover, the method enables the partial reconstruction of the original content in missing portions previously removed by cropping.

A common image counterfeiting technique involves concealing undesired elements in the portrayed scene with a portion of pixels copied from the same image. In this work, a method is proposed to detect duplicated regions, even when the copied part has undergone post-processing operations, such as reflection, rotation, scaling or/and JPEG compression. This is efficiently achieved, in terms of memory usage and computational cost, by mapping overlapping blocks of pixels to colour-dependent feature vectors and one-dimensional (1-D) descriptors, derived from log-polar maps.

To my beloved wife

Maria Elena.

Acknowledgements

My time as student has been touched in many ways by a number of excellent people, whose enthusiasm and diligence remains in my mind as an invaluable source of inspiration. First, I would like to thank my supervisor, Professor Asoke K. Nandi, for his guidance and support throughout the development of this project. My gratitude also goes to Dr. Lu Gan for her encouragement and advice during my first year.

I want to thank the past members of the SCP group, especially to Dr. Sonu Punoose, Dr. Luciano Sarperi and Dr. Alfonso Rojas-Dominguez, for their friendship and guidance in the administration of the system at the early stages of my project. Thanks to Dr. Sameh Salem, Dr. Nancy M. Salem and Dr. Tingting Mu for all their patience, kindness and advice, as well as Obilor, Dr. Jingbo Gao, Dr. Abrar Shafayat, Waqar, Hussain and Nazmat for making my time in the group such an enjoyable experience. I feel particularly lucky that, during these years, I had the opportunity of sharing the room with Obilor, whose joyful company, "operatic" singing talent and critical thinking made the lab a very pleasant place to be. I will always remember all those interesting conversations that took place in the lab.

I owe special thanks to the National Science of Science and Technology (CONACyT) of Mexico for the financial support for this research.

I reserve my deepest acknowledgements to the dedicate of this thesis, my beloved wife Maria Elena, for all her love, patience, understanding, support and encouragement, as well as for the countless special moments that we have gone through. Thanks for making this adventure such an agreeable journey *SMN*.

I want to express my gratitude to my parents and my sister for their unconditional support, encouragement and love throughout my life. I will always treasure the strong principles of love, loyalty and ethics that has governed in my parent's matrimony. I also acknowledge the support provided by my wife's family from the beginning of this experience.

I am very grateful to the enthusiastic Mexican community in Liverpool that has made these years memorable. I wish I could mention them all individually. I want to thank Elda, Carlos, Mario, Erika, Lindsey and Antoinette for their invaluable friendship, advice and support.

Thanks to Lindsey and Dr. Sonu Punoose for their valuable comments and suggestions during the proofreading of this thesis.

Contents

List of Figures

List of Tables

Acronyms

1-D	One-Dimensional
ACC	Accuracy
AWGN	Additive White Gaussian Noise
BAM	Block Artifact Measure
BMP	Bitmap
BPPM	Block Posterior Probability Map
CCD	Charge-Coupled Device
CFA	Colour Filter Array
CRF	Camera Response Function
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
ECC	Error Correction Code
EM	Expectation-Maximisation
FFT	Fast Fourier Transform
FMT	Fourier-Mellin Transform
FPR	False Positive Rate
HVS	Human Visual System
IDWT	Integer Discrete Wavelet Transform
IQA	Image Quality Assessment
JPEG	Joint Photography Experts Group
LCA	Lateral Chromatic Aberration

LPM	Log-Polar Mapping
LSB	Least Significant Bit
LSBP	Least Significant Bit-Plane
LUT	Look-Up Table
MAC	Message Authentication Code
MD5	Message Diggest (Algorithm) 5
MSB	Most Significant Bit
MSBP	Most Significant Bit-Plane
MSE	Mean Squared Error
PCA	Principal Component Analysis
PNG	Portable Network Graphics
PSNR	Peak Signal-to-Noise Ratio
RGB	Red, Green and Blue
SHA	Secure Hash Algorithm
SIFT	Scale Invariant Feature Transform
SSIM	Structural Similarities
SURF	Speed-Up Robust Features
SVM	Support Vector Machine
TIFF	Tagged Image File Format
TNR	True Negative Rate
TPR	True Positive Rate
VQ	Vector Quantisation
XOR	Exclusive OR
YCbCr	Luminance/Chrominance colour space

Chapter 1

Introduction

As technology advances, image editing systems are equipped with increasingly powerful and sophisticated processing tools that enables the creation of convincing image forgeries, which cannot be easily identified, even by trained observers. For some applications, certain digital manipulations may serve legitimate purposes, such as photograph enhancement or lossy compression. However, in other cases, the manipulations, whether malicious or not, could negatively influence the interpretation of the visual information. This has motivated the development of technology that can deliver certainty about the integrity of digital images.

Table 1.1 presents a general classification of the techniques conceived to address this concern. Friedman [1] introduced the concept of the Trustworthy Digital Camera, whereby a digital signature, generated within the camera at the time of capturing, is appended to the image file as metadata. This technique is *active*, in the sense that the signature is deliberately generated for the purpose of integrity verification, and *non-invasive*, as the image is not altered at all by this process. The main limitation of this approach stems from its dependence to the image file format used. This is because the metadata could be destroyed if the image is converted to a different format, even if its content remains intact.

Table 1.1: Classification of techniques for integrity verification of images.

	Active	**Passive**
Non-invasive	Digital signatures	Image forensics
Invasive	Digital watermarking	-

Digital watermarking[1] is another active approach that describes mechanisms to embed imperceptible information in the image itself. To provide integrity verification, a typical watermarking method splits the image into two parts: one part comprised of the most relevant perceptual information and another part, which can be distorted without significantly altering the image content. In the watermarking process, the part with less perceptual importance is replaced with a message authentication code (MAC) derived from the relevant part involving some security parameters – e.g. a secret key. When the receiver employs the correct security parameters, the originally hidden MAC can be retrieved and compared with a second MAC derived from the relevant part of the host[2] image. If these are identical, the system assumes that the host image has not been altered. This approach is *invasive* in the sense that the image is almost inevitably distorted by the watermarking process.

Watermarking techniques that serve the purpose of integrity verification can be divided into *fragile* and *semi-fragile*. A fragile watermarking scheme is designed to determine whether or not a host image has been distorted at all. That is, an image is deemed altered even if only one bit has been corrupted. Yet, since the watermark is hidden in the content of the image, independently of the storing file format, a host image can be converted to a number of image file formats that do not introduce any distortion, such as BMP, PNG, TIFF, etc. Semi-fragile watermarking methods, on the other hand, adopt relaxed integrity criteria, which tolerate distortions that make no "significant" changes to the semantic meaning or perceptual quality of the content (e.g. lossy compression), while still rejecting malicious manipulations. However, the boundary between legitimate and illegitimate distortions is not always clear-cut and

[1] Hereinafter, the term *digital* will be omitted when referring to *digital watermarking.*

[2] Throughout this work, the terms *host image* and *watermarked image* will be used interchangeably.

generally varies from application to application.

Since, ideally, the authentication information should be generated at the time of capturing, in practice, the use of active techniques (i.e. digital signatures and watermarks) is reserved for controlled environments with properly equipped capturing devices, such as surveillance applications or military systems. Image forensics describe *passive* techniques aimed at identifying possible traces of tampering in the absence of any type of authentication information generated in advance. This is typically achieved by detecting anomalies in the intrinsic statistics of digital images that may suggest, in some cases very strongly, that an image has been manipulated. The main challenges of this non-invasive approach are posed by the plethora of possible manipulations an image may go through, including sophisticated attacks designed to deceive existing forensic methods.

1.1 Motivations

The work of this thesis focuses on fragile watermarking and image forensics for the purpose of integrity verification of digital images.

Fragile watermarking

The clear integrity criteria used in fragile watermarking provide a solid and reliable framework, which facilitates functionalities that are worthy of investigation.

A highly desirable functionality, termed *tampering localisation* (or simply *localisation*), consists in identifying altered regions of a host image [2]. There are some potential security threats relevant to localisation [3, 4], which have to be considered to develop reliable schemes. Exploring alternative approaches to verification could be beneficial in terms of security and localisation accuracy.

Another very desirable functionality, commonly referred to as *self-recovery*, consists in partially reconstructing the original content in distorted regions [2]. Unfortunately, the embedding distortion introduced by most of the existing schemes would make them unsuitable for some applications. Furthermore, the self-recovery capabilities of most of the existing methods can be seriously affected by cropping (cutting out and removing a portion of the image). These shortcomings could be effectively addressed by combining two different watermarking mechanisms in a single method.

Image forensics

Image forensics, on the other hand, is a promising emerging field with many challenges ahead. Because of the complex nature of the addressed problems, forensic evidence is usually derived from a diverse set of mechanisms, rather than a single algorithm [5].

A common manipulation consists in covering undesired people or objects with a region of pixels copied from the same image. This manipulation is usually referred as to *copy-move*. To perform an efficient search, this problem is typically addressed by mapping overlapping blocks of pixels to dimension-reduced feature vectors. However, most of the existing methods fail when the replicated area has undergone geometric changes, such as rotation or scaling. Using dimension-reduced descriptors, derived from log-polar maps, could be used to detect duplicates, even when the copied region has been reflected, rotated, or/and scaled.

1.2 Contributions

The original contributions of this thesis are summarised below.

- A fragile watermarking method that takes another approach to verification, wherein non-overlapping blocks of pixels are not authenticated independently of each other, is proposed. Instead, the authentication information retrieved from every block is examined to identify significant similarities, which are then used to determine whether or not the image has been watermarked and localise altered pixel blocks. (Chapter 4)

- A fragile watermarking method with improved tampering localisation and exact self-recovery capabilities is presented. The parameters of the scheme can be readily adjusted to control the trade-off between the embedding distortion and the localisation/self-recovery performance of the system. Furthermore, the method provides resilience to cropping, thereby enabling the restoration of pixels in missing portions of the image removed by cropping. It is important to mention that this is the first attempt to address the problem of recovering pixels from cropped images. (Chapter 5)

- An experiment is conducted to investigate to what extent the localisation/self-recovery performance of the proposed fragile watermarking method is affected by the size of the tampered region and the proportion of altered pixels, in comparison with an existing scheme. (Chapter 5)

- A forensic method is proposed to detect duplicated regions of pixels, even when the copied portion has been subjected to reflection, scaling or/and rotation, as well as JPEG compression. This problem had not been comprehensively addressed in today's literature. (Chapter 6)

1.3 List of Publications

The following, are publications derived from the work of this thesis.

Accepted

- S. Bravo-Solorio, and A. K. Nandi, "Secure fragile watermarking method for image authentication with improved tampering localisation and self-recovery capabilities," to appear in *Signal Processing*, doi:10.1016/j.sigpro.2010.07.019.

- S. Bravo-Solorio, L. Gan, A. K. Nandi, and M. F. Aburdene, "Secure private fragile watermarking scheme with improved tampering localisation accuracy," in *IET Information Security*, vol. 4, no. 3, pp. 137-148, 2010.
- S. Bravo-Solorio, and A. K. Nandi, "Fragile watermarking with improved tampering localisation and self-recovery capabilities," in *Proc. of EUSIPCO – European Signal Processing Conference*, Aalborg, Denmark, 2010.
- S. Bravo-Solorio, and A. K. Nandi, "Exposing duplicated regions altered by reflection, rotation and scaling through image forensics," in *Proc. of Mathematics in Defence*, Farnborough, Hampshire, England, 2009.
- S. Bravo-Solorio, and A. K. Nandi, "Passive forensic method for detecting duplicated regions affected by reflection, rotation and scaling," in *Proc. of EUSIPCO – European Signal Processing Conference*, Glasgow, Scotland, 2009.
- S. Bravo-Solorio, L. Gan, A. K. Nandi, and M. F. Aburdene, "Fragile logo watermarking for public authentication," in *Proc. of ICASSP – IEEE International Conference on Acoustics, Speech and Signal Processing*, Las Vegas, NA, USA, 2008.

Under review

- S. Bravo-Solorio, and A. K. Nandi, Automated detection and localisation of duplicated regions affected by reflection, rotation and scaling in image forensics, under review in *Signal Processing*, (submitted in April, 2010).

1.4 Thesis Overview

This thesis is organised as follows:

Chapter 2 describes the general requirements of fragile watermarking, with an emphasis on tampering localisation and its associated security risks, including a brief account of some important attacks. This is followed by a review of fragile watermarking methods developed to meet the requirements of different problem domains.

Chapter 3 introduces the basic principles of image forensics for the purpose of tampering detection. In addition, this chapter surveys the diverse range of existing image forensic tools, their potential limitations and some possible countermeasures that have been developed to hinder their detection capabilities.

Chapter 4 revisits an existing fragile watermarking method, whereby the same key can be used to watermark a series of images, each one associated to a unique index. An analysis is presented of the security limitations of this scheme, particularly when higher tampering localisation is required. To overcome these shortcomings, a method is proposed, whereby the authentication data retrieved from non-overlapping blocks of pixels is examined to determine, almost unequivocally, whether the image has been watermarked and localise possible tampered blocks. Experimental results are presented to demonstrate the effectiveness of the proposed scheme.

Chapter 5 presents a fragile watermarking method aimed at providing improved localisation and exact self-recovery capabilities. The proposed algorithm combines a mechanism resilient to cropping, which detects distorted blocks of pixels, with a second mechanism that refines the localisation and estimates the original watermarked pixels in altered regions. An extensive investigation is presented into how the size of the tampered region and the proportion of altered pixels affect the performance of the proposed method, in comparison with a state-of-the-art scheme.

Chapter 6 presents a forensic method aimed at detecting duplicated regions of pixels, even when the replicated portion has undergone geometric changes, specifically reflection, rotation, or/and scaling. The proposed algorithm maps overlapping blocks of pixels to colour-based features to constrain the search for duplicates, thereby reducing the computational cost. Additionally, to perform an efficient search, every pixel block is mapped to reflection/rotation-invariant 1-D descriptors derived from log-polar maps. Extensive experimental results, including a comparative evaluation with an existing method, are presented to demonstrate the effectiveness of the proposed method.

Chapter 7 concludes this thesis and describes various possible directions for future research following on from this work.

Chapter 2

Fragile Watermarking

2.1 Introduction

This chapter presents a brief review of fragile watermarking. Robust and semi-fragile watermarking are beyond the scope of this work. Readers interested in digital watermarking from a more general perspective are referred to [6] and [2].

The chapter is structured as follows. Fragile watermarking methods can not only identify changes in a host image, but also provide additional functionalities discussed in Section 2.2. The common requirements that have to be observed in fragile watermarking are described in Section 2.3. Section 2.4 focuses on some sophisticated attacks, which pose a serious threat to some fragile watermarking methods. This is followed by a brief literature review on fragile watermarking in Section 2.5. Finally, Section 2.6 presents the summary of the chapter, along with some discussion.

2.2 Further Functionalities

Fragile watermarking systems can produce binary answers (i.e. yes/no) to state whether or not the host image has been changed. However, the fact that the watermark undergoes the same transformation as the host image opens up the possibility of providing additional capabilities, for example [2, 7]:

- *Tampering localisation* – Regions that have been distorted are identified, while verifying that the remainder of the image is intact. Thus, even in presence of corrupted images, one could still rely on the content in regions that have been reckoned to be genuine. Moreover, by being aware of the tampered regions, one may infer the purpose of the tampering or identify possible candidate perpetrators [2].

- *Reversible embedding* – To embed the authentication information, the host image is modified, thereby introducing some amount of distortions. In many applications, distortions induced can be accepted, as long as the relevant perceptual information is preserved. However, even imperceptible alterations in the pixels are undesirable in fields that handle very sensitive information, such as in military or medical application domains [8, 9]. To address this concern, some methods embed the authentication data in an invertible manner, so that the host image can be restored to its original state, before the embedding took place. In literature, these methods are referred to as *reversible*, *invertible*, *lossless* or *distortion-free* methods.

- *Self-recovery* – Once an image is reckoned to have been altered, it would be highly desirable to restore the content to its original state, before the manipulation occurred. Typically, methods that afford this functionality embed the authentication data together with additional information, which can be used to restore the image [10–12].

2.3 General Requirements

To provide integrity verification, the requirements that watermarking systems have to comply with are described below.

2.3.1 Imperceptibility

Independently of the particular applications, the perceptual transparency of the hidden data might be one of the most important requirements for watermarking. Affecting the quality of the image severely may render the host image useless for the intended purposes. Some properties of the human visual system (HVS) have been studied to optimise the energy of the watermark without exceeding the perceptible threshold [13, 14]. Nonetheless, this practice is uncommon in literature on fragile watermarking, as most of the schemes work under the assumption that the embedded information should be rather weak. Instead, quantitative distortion metrics are commonly provided to compare the perceptibility of the watermarks embedded by different methods. Nowadays, the most popular distortion measure in the field is the peak signal-to-noise-ratio (PSNR), measured in decibels. The PSNR between two images, say X and Y, both sized $n_1 \times n_2$, is typically defined as [15],

$$\mathrm{PSNR} = 10 \log_{10} \frac{\mathrm{max}_X^2}{\mathrm{MSE}} \ , \tag{2.1}$$

where max_X is the maximum possible pixel value in X (usually 255 in 8-bit grey-scale images) and MSE stands for the mean squared error, given by,

$$\mathrm{MSE} = \frac{1}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} (X(i,j) - Y(i,j))^2 \ . \tag{2.2}$$

It is well-known that this measure does not emulate the response of the HVS. Even though some efforts have been made to construct distortion metrics that correlate well with the HVS, such as the image quality assessment (IQA) and the structural similarities (SSIM) [16–19], none of these metrics has been widely adopted in watermarking literature.

2.3.2 Security

To develop effective security mechanisms, it is important to understand the potential goals of attackers [20, 21]:

- *Undetected distortions* – The aim is to manipulate a host image without being detected by the verification system.
- *Unauthorised embedding* – Here, the idea is to embed a legitimate watermark in an image, without the consent of the watermark owner.
- *Information leakage* – The goal is to obtain some information about the security-related parameters of the algorithm, such as the secret key.

Additionally, the security measures of each scheme should be designed considering the potential scenarios in a particular application domain. Some of the possible scenarios include the following [20]:

- *Host image* – The attacker has access to a single host image.
- *Multiple host images* – Various host images are available to the attacker.
- *Verification device* – The attacker has unlimited access to the verification system. Thus, different versions of the forgery can be exhaustively submitted to the system trying to find one that is deemed authentic (oracle attack [22]).
- *Original image* – Multiple pairs of original/watermarked images are available for the attacker.
- *Embedding system* – The embedding system is available for the attacker, who can submit her/his own images trying to find possible security flaws in the system.

2.3.3 Computational cost

Since, ideally, a digital image should be watermarked at the time of capturing [20], the embedding algorithm should be hard-wired in the camera. Therefore, it is important to keep the computational cost of the embedding system as low as possible. On the other hand, the importance of the computational cost of the verification system, which could be installed in a personal computer, depends on the workload of the application. For example, consider a system used to verify the integrity of CCTV images submitted as evidence in court. In this case, the response time of the verification system does not seem to be of paramount importance. However, an on-line system aimed to verify the integrity of thousands of images on a daily basis should be designed under stricter constraints on response time.

2.4 Security Risks Associated with Localisation

There are some security risks that must be considered when providing localisation. In particular, some attacks, commonly known as vector quantisation (VQ) attacks, have been proposed to create counterfeit images that would be regarded as authentic and legitimate by vulnerable systems [3]. This attack exploits block/pixel-wise independence to generate VQ code-books from a set of images watermarked with the same key. The code-books can be used to embed a valid watermark in an illegitimate image without the consent of the watermark owner.

To elaborate on this issue, consider a simplified version of a method vulnerable to this attack [23], whereby the input image is first divided into non-overlapping blocks of pixels. Then, the seven most significant bit-planes (MSBPs) in each block are independently hashed to generate a MAC, which is subsequently inserted in the least significant bit-plane (LSBP) of the same block. Observe that the MAC inserted in each block only depends on the content of the block itself. Therefore, swapped blocks would go unnoticed by the verification system. Even worse is the fact that a completely new image could be assembled with independent blocks if the adversary has access to a set of host images, all watermarked with the same key.

To thwart VQ attacks, it is necessary to establish some type of pixel/block-wise dependence [3]. The mechanisms that have been proposed to achieve this goal can be classified in three groups:

Type 1: *Deterministic dependence* – The data embedded in each pixel/block implies information concerning one or more contiguous pixels/blocks, together with the pixel/block itself. This type of dependence is deterministic and can be easily discerned from the very algorithm.

Type 2: *Non-deterministic dependence* – The data generated from the content of each pixel/block is inserted into another pixel/block selected pseudo-randomly.

Type 3: *External information-based dependence* – The data embedded in each pixel/ block involves information that cannot be deduced from the image itself – e.g. an image index.

These mechanisms can effectively prevent VQ attacks. However, Barreto *et al.* [4] noticed that some deterministic mechanisms (Type 1) are vulnerable to what they call transplantation, or copy-and-paste, attacks. To illustrate the principles of this attack, consider the scenario depicted in Figure 2.1. For the sake of simplicity, assume that the MAC embedded in each block of the host image depends on the context of its left-neighbour block, together with the context of the block itself. Observe that the circle can be duplicated in another part of the image as long as the context of the blocks at both sides of the dashed area is preserved. Even though the content of the image has changed significantly, the counterfeit would be regarded as authentic. The same results would be obtained with a region copied from another host image watermarked with the same key. Images with large areas of uniform pixel values are potential targets of this attack.

During the course of this investigation, it was observed that the abilities of some existing watermarking methods to localise copy-and-paste attacks vary significantly depending on whether the supplanting region was copied from the same or a different image. Hence, in this work, these manipulations will be considered as separate

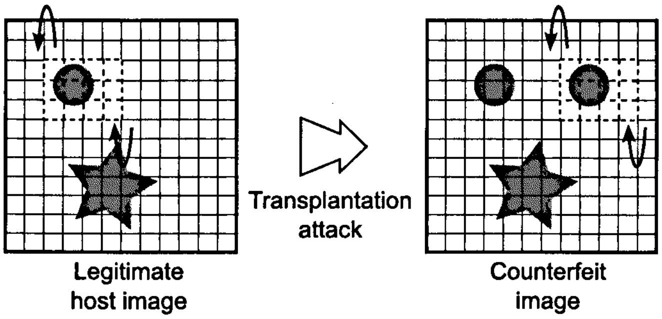


Figure 2.1: Illustration of the copy-and-paste attack.

attacks, referred to as *inner-copy-and-paste attacks* and *outer-copy-and-paste attacks*, respectively.

2.5 Literature Review

This section presents a concise review of fragile watermarking, also referred to as *exact* or *hard* authentication [2, 21]. First, the most successful mechanisms designed to provide localisation, while effectively thwarting VQ attacks, are described. The impact that these methods have on the quality of the host images is minor, as the information is only embedded in the LSBP of the image. Methods that require higher localisation, usually at the expense of increasing the embedding distortion, are reviewed next. This is followed with a description of distortion-free and self-recovery schemes.

2.5.1 Tampering localisation with minimum embedding distortion

Wong and Memon [24] proposed a block-wise method, whereby a MAC is independently embedded in non-overlapping blocks of 12 × 12 pixels. The MAC depends on the context of the block, a secret key, an image index and a block index. This mechanism can successfully prevent VQ attacks and can be readily adapted to afford public key authentication. However, the system has two major limitations: 1) The correct image index has to be known to the receiver. Such a requirement might be impractical

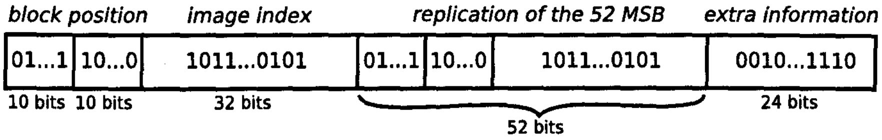


Figure 2.2: Authentication data structured for each block in Fridrich's approach.

for many applications. 2) The logo is retrieved for visual inspection, but automated detection is possible only if the logo is provided to the verification system.

Wong and Memon's work has been extended in various directions. For example, the practical aspects of standard asymmetric-key encryption algorithms have been studied in [25] to afford real public authentication. In [26], the reliance on the image index is replaced with a master/session key-based mechanism used to initialise the encoding system. At the receiver end, the watermark can be retrieved only if the system is provided with the correct pair of keys. Thus, the transmitter and the receiver could confidentially agree on a master key, whereas a new session key could be conveyed through an insecure channel, before transmitting a single or a sequence of host images.

In Fridrich's approach [20], the information about the image is separated from the authentication data. Non-overlapping blocks of 16 × 8 pixels are independently hashed and then XOR-ed with the structure illustrated in Figure 2.2. The resulting bit strings are encrypted, using either a symmetric or an asymmetric algorithm, and embedded in the LSBP of each block. At the receiver side, a block is deemed authentic if the 52 most significant bits (MBSs) of the retrieved structure are identical to its subsequent 52 bits. Additionally, the bit strings extracted from all the blocks can be further analysed to identify swapped blocks and/or blocks copied from other images. This method manages to overcome the two limitations identified in Wong and Memon's scheme. However, during the course of this investigation, it was found that brute-force attacks may pose a serious threat to this method in applications that need to use smaller block-sizes to achieve higher localisation.

Celik *et al.* [27, 28] proposed a method where non-overlapping blocks of 13 × 8 pixels

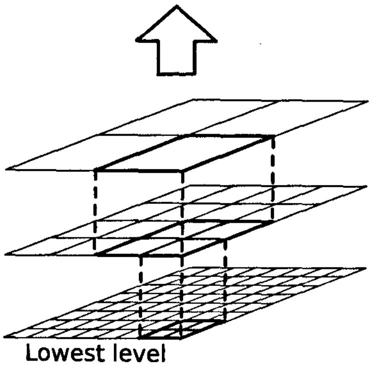


Figure 2.3: Hierarchical structure proposed by Celik *et al.*.

represent the lowest level in the hierarchical structure illustrated in Figure 2.3. Every block in upper hierarchical levels is assembled as 2 × 2 adjacent blocks of the preceding level. Then, a MAC is computed for every block in all the hierarchical levels. The MACs are encrypted and embedded in the LSBP of the image in accordance with the hierarchical structure, so that every block in the lowest level will carry portions of the MACs obtained from its associated blocks in upper levels, together with its own MAC. In essence, the lowest level is intended for localisation, whereas the remaining levels are used to thwart VQ attacks. Another hierarchical approach was presented in [29]. In this case, a sparse set of wavelet coefficients are watermarked in accordance with a contextual non-deterministic dependence mechanism, which involves all the wavelet coefficients across the wavelet decomposition levels. The aim is to protect all the pixels without altering all the wavelet coefficients. This way, the embedding distortion is reduced without compromising the security of the method.

In [30], every wavelet coefficient in the coarser sub-band, at the first decomposition level, is independently mapped to a 4-bit code, by means of a non-uniform pseudo-random quantisation. Each code is embedded in the LSBP of the block of 2 × 2 pixels associated to another coefficient selected pseudo-randomly. At the receiver side, a bitmap is encoded to localise the coefficients where the computed codes fail to match the extracted ones. Then, a post-processing mechanism is computed on the bitmap to remove isolated spots and improve the validation of genuine content. The scheme is capable of identifying whether an attack has been performed on the image or on the watermark itself.

2.5.2 Higher localisation accuracy

Potential benefits can be derived from fragile watermarking methods that afford higher localisation capabilities, without compromising their security. For example, consider the case of a host image affected by salt-and-pepper noise. Identifying the location of the altered pixels with pinpoint accuracy can be useful to determine whether the semantic meaning of the content has been affected significantly [31]. Furthermore, depending on the nature of the distortion, it might be possible to restore the altered content by means of denoising or inpainting techniques [32–35].

In Yeung and Mintzer's method [36], a secret key is used to generate a look-up table (LUT) that maps each pixel intensity value to a binary value. The intensity value of every pixel is adjusted in accordance with the LUT, such that the altered pixel maps to the corresponding binary value in a secret logo. The method, whose security relies on the secrecy of the key and the logo, manages to localise genuine pixels very accurately. However, two important limitations have been found in the scheme: 1) only half of the tampered pixels, on average, are effectively identified; and 2) if the logo is known, the LUT can be readily deduced [3]. In fact, on average, only two images, watermarked with the same key and logo, are necessary to infer 90% of the LUT [37, 38].

To cope with the second limitation in Yeung and Mintzer's scheme, the LUT can be replaced with a standard encryption algorithm, while embedding an additional image index in non-overlapping pixel blocks [39]. This is insufficient, however, when the attacker has unlimited access to the verification system. Alternative methods have been explored in [40, 41], where watermarks are modulated in accordance with either two LUTs or chaotic patterns. Nevertheless, the original logo has to be known by the verification systems. This requirement may be impractical for some applications. The mechanisms in [39–41] manage to validate the authentic pixels very accurately, while effectively preventing VQ attacks. However, they fail to localise, on average, half of the altered pixels.

Zhang and Wang [31] showed that the localisation can be enhanced at the expense of increasing the embedding distortion. The MAC, derived from the 5 MSBs of every single pixel and a pseudo-random code, is embedded into the 3 LSBPs of the image. At the receiver end, two distributions, corresponding to the altered and genuine pixels, are used to localise pixels corrupted in their 5 MSBs. If the tampered region is not too extensive, the method manages to localise altered pixels very accurately. Moreover, this system can effectively localise scattered altered pixels, such as salt-and-pepper noise.

In He *et al.*'s method [42], a pseudo-random circular block-mapping sequence is first generated with a secret key. The MAC computed for a block is embedded in the subsequent block in the sequence. One of the 3 least significant bits (LSBs) in every pixel is pseudo-randomly selected to carry the watermark. The authors demonstrate that the scheme is secure, even when using a block-size of 4 × 4. The localisation performance is consistently high, as long as the tampered region does not extend beyond 50% of the number of pixels in the host image. Nevertheless, this method fails to localise scattered altered pixels.

2.5.3 Reversible embedding

The need for reversible embedding stems from the fact that, for some application domains, preserving the quality of an image is critical for its intended purposes. For example, perceptible distortions of medical images may lead to misdiagnoses. In fact, even if the content has not been perceptibly changed, any distortion inflicted to a medical image might be considered in a malpractice law suit [2, 9]. In other fields of high precision nature, such as remote sensing or military applications, it is desirable that the original content can be recovered. The four basic design principles that have been developed for the purposes of lossless embedding are described below. It is important to mention that, whatever the employed technique may be, preserving the quality of the host images is still a critical requirement.

Modular arithmetic

In [43], an authentication signal is embedded in the image by using modulo-256 addition. At the receiver end, after extracting the authentication data, the embedding distortion can be removed by using modulo-256 subtraction, thereby restoring the image to its original state before the watermarking process. The modulo addition enables reversibility, but may introduce perceptible artifacts in the form of salt-and-pepper noise, as a result of pixel values close to zero being flipped to values close to 255 and vice versa. An extension of this method was presented in [44].

Lossless compression of bit-planes

In [44, 45], the lower bit-plane with sufficient redundancy is losslessly compressed and subsequently reinserted in the same bit-plane, together with a MAC derived from the original content. This way, the original image can be restored at the receiver end. In [46, 47], the compression algorithm uses the non-distorted part of the host image, as side-information, to maximise the embedding capacity. This mechanism has been extended in [48, 49] to provide public distortion-free authentication with localisation capabilities. Xuan *et al.* [50, 51] proposed to select the bit-plane from the high frequency sub-bands of the integer discrete wavelet transform (IDWT) domain, which commonly have high redundancy, thereby leading to higher compression rates.

Pixel groups flipping

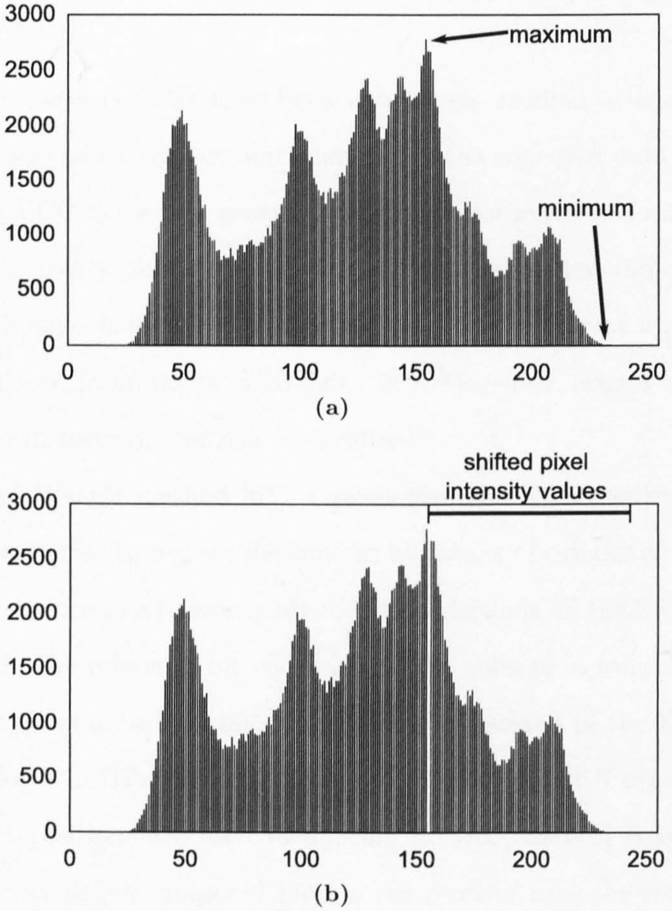
In Fridrich *et al.*'s scheme [52], non-overlapping groups of adjacent pixels are classified according to some uniformity criteria as: (R) regular, (S) singular, or (U) unusable. An invertible function $\mathcal{F}$ is defined to map groups in the category (R) to the category (S) and vice versa; groups in the category (U) do not change. An example of a simple invertible function $\mathcal{F}$ can be the one that flips the LSB of an 8-bit grey-scale pixel, i.e. $0 \leftrightarrow 1, 2 \leftrightarrow 3, \ldots, 254 \leftrightarrow 255$. During embedding, each group in the categories (R) and (S) are used allocate a bit of information; (R) $= 0$ and (S) $= 1$. So, the function $\mathcal{F}$ is used to flip the groups that not match the to-be-embedded bit. To enable the removal of the embedding distortion, a bitmap that captures the initial state of the groups (R) and (S) is losslessly compressed and embedded along with the rest of the payload.

Difference expansion

Tian [53] presented a high capacity method, which exploits the fact that, in typical natural images, adjacent pixels have similar intensity values. The central idea is to expand the difference between pairs of adjacent pixels, in an invertible way, to allocate a bit of information in every pixel-pair. The mechanism is based on the integer Haar wavelet transform, which can lead to underflow/overflow in some pixels. To avoid this problem, only pixels whose difference can be extended without causing overflow/underflow are used and their location is recorded in a bitmap, which is losslessly compressed and embedded together with the rest of the data. Comparison results reported in [53] show that this method achieves higher embedding capacity than the schemes in [46, 47, 52]. Furthermore, Tian's approach has been extended for using triplets and quads of pixels to improve the embedding capacity [54–58].

Histogram shifting

The mechanism detailed in [59, 60] starts with the identification of the most and the least frequent pixel values in the input image. As an example, both pixel values are illustrated in the histogram of Figure 2.4(a), using the standard 512 × 512 grey-scale version of Lena. Then, the histogram is shifted 1 unit towards the minimum value, as illustrated in 2.4(b). The value of the most frequent pixel value is adjusted to encode every bit in the data payload. If the minimum value in the histogram is non-zero, a gap is deliberately produced and the locations of the altered pixels are appended to the



(a)

(b)

Figure 2.4: Illustration of histogram-shifting. (a) Histogram of the grey-scale 512 × 512 version of Lena. (b) Shifted histogram.

embedded data payload. The authors show that the embedding distortion produced by this method is significantly lower than in [45, 46, 50]. This approach has also been extended to increase the embedding capacity [61, 62], and provide localisation capabilities [63].

2.5.4 Self-recovery

Restoring the content of the altered regions is another very desirable functionality. Below are reviewed some of the most relevant mechanisms that have been proposed to reconstruct, either exactly or approximately, the original content of a host image.

Exact restoration

Error correction codes (ECCs) have been extensively studied in communications to detect and, in many cases, correct perturbations in the conveyed data [64]. In [10, 11], a Reed-Solomon ECC is used to generate parity bits for every row and column of the input image. The parity pixels are encrypted with a secret key and embedded in the 2 LSBPs of the image. It is reported that this method can correct up to 13 bytes in a single column or row, in an image sized 229 × 299. Moreover, even if the pixels cannot be corrected, the distortions can still be localised.

In Zhang and Wang's method [65], a pixel-wise and a block-wise mechanism are hierarchically structured to restore the content by means of exhaustive attempts. Each pixel is used to produce to a reference bit code that depends on the 5 MSBs of the pixel and a secret key. The reference bit codes of disjoint subsets of roughly 16 pixels each are combined to generate a long bit string, which is inserted in the 160 bit positions, selected from the 3 LSBPs of non-overlapping blocks of 8 × 8 pixels. Finally, the remaining 32 bit positions are used to allocate a MAC, derived from the content of the block. After localising tampered blocks, the receiver uses the retrieved reference bit codes to enhance the localisation and exhaustively calculate the original 5 MSBs of distorted pixels. Nonetheless, the system is insecure against VQ attacks, as the MAC encoded by the block-wise method depends only upon the content of the block itself (recall Section 2.4). This method has been extended, in [66], to estimate approximate pixel values when the exhaustive mechanism failed to restore their value exactly. Thus, smaller altered areas can be perfectly reconstructed, whereas larger tampered regions can be restored only approximately. Furthermore, the authentication data encoded by the block-wise method includes information about the image to prevent VQ attacks. However, in [65, 66], the watermark is always embedded in the 3 LSBPs of the image. Such a lack of flexibility may limit their use in applications where less embedding distortion is needed.

Zhang and Wang [67] proposed a method that elegantly embodies lossless embedding and restoration capabilities. Here, a watermark comprised of reference bits and check bits is embedded in non-overlapping blocks of 8 × 8 pixels by using the difference expansion technique described in Section 2.5.3. The receiver extracts the watermark and uses the check bits to localise the tampered blocks by means of a statistical mechanism. Finally, a binary linear system, which involves the retrieved reference bits, is solved to calculate the value of altered pixels. This is the only existing method that is capable of reconstructing the original non-watermarked image out from an altered host image. In practice, however, the content cannot be restored when the tampered area is larger than 3.2% of the image.

Approximate restoration

In Fridrich and Goljan's scheme [12], a JPEG compressed version of the image, produced at 50% quality factor, is encrypted and embedded in the LSBP of non-overlapping blocks of 8 × 8 pixels. The extracted low-quality image is used at the receiver side to locate and recover corrupted blocks. Alternatively, to improve the quality of the restored image, a version of the image compressed at a higher quality factor can be embedded in the 2 LSBPs of the pixel blocks.

In the method proposed by Lin *et al.* [68], non-overlapping blocks of 4 × 4 pixels are split into four 2 × 2 sub-blocks. Following on from the generation of a secret block-mapping sequence, the average intensities of the four sub-blocks, as well as some parity-check bits, are embedded in the 2 LSBPs of the sub-blocks in the block that comes next in the sequence. The verification side uses a hierarchical mechanism to localise and recover the altered blocks. However, without the secret key, an adversary could estimate the correlation between blocks to generate counterfeits that would go unnoticed by the verification system [69].

A self-recovery watermarking method for 24-bit RGB-colour images was proposed by Wang and Chen [70]. In this scheme, a reference bit code, comprised of authentication and recovery information, is encoded for non-overlapping blocks of 2 × 2 pixels.

The authentication information is derived from the content of the block and some general details about the image, while the recovery information is formed by the averaged colour components in the block. The receiver implements a majority-voting mechanism to localise altered blocks, which can be reconstructed by using the recovery information. The method seems to outperform Lin *et al.*'s scheme in terms of quality of the restored images. Nevertheless, the lack of adequate security mechanisms has been exploited in [71] to mount successful counterfeiting attacks.

In Lee and Lin's method [72], a secret key is used to produce a sequence of pairs of non-overlapping blocks of 2 × 2 pixels. The average intensities of a block pair, as well as some parity-check bits, are inserted in the 3 LSBPs of the block pair that appears next in the sequence; the two blocks carry the same information. The receiver side implements a hierarchical localisation and recovery mechanism, as in Lin *et al.*'s method. Then, a post-processing mechanism is used to fill the blocks that could not be recovered. However, reported results show that all the blocks could be reconstructed without using the post-processing mechanism, even when the altered area extended to up to 50% of the host image. Even though the quality of the recovered content drops for larger areas, some dim elements can still be recognised. Unfortunately, the method inherited the aforementioned security problems in Lin *et al.*'s method.

Zhang *et al.* [73] proposed a variation of the method in [67] to afford approximate restoration. In this case, non-overlapping blocks of 8 × 8 pixels are independently transformed to the discrete cosine transform (DCT) domain. A sequence of bits, comprised of the quantised lowest coefficients of each block, is spread over 160 bit locations pseudo-randomly selected from the 3 LSBPs in every pixel block. The remaining 32 bit locations in each block are used to allocate a MAC derived from the content of the block and a block index. The receiver detects the tampered blocks and uses the bits retrieved from the genuine blocks to form a binary linear equation system. If the number of bits retrieved from genuine blocks is sufficiently large, the equation system can be solved to restore the original sequence of bits, thereby reconstructing the lowest DCT coefficients of the tampered blocks. Reported theoretical estimations suggest that the sequence of bits can be perfectly recovered when tampered area extends to less than 59% of the host image.

2.6 Summary and Discussion

Fragile watermarking describes mechanisms aimed at offering certainty about the integrity of digital images. Two basic requirements have to be observed when designing a fragile watermarking method. First, the embedded watermark has to be imperceptible. Second, adequate security mechanisms have to be considered to prevent malicious attacks, generally intended to generate forgeries that would go unnoticed by the verification system. The security mechanisms have to be tested in different scenarios, e.g. in cases where the adversary has unlimited access to the verification system.

Localising tampered areas, while verifying the integrity of the remainder portion of the host image, is a highly desirable functionality in fragile watermarking. Nonetheless, methods with localisation capabilities need to a establish some type of pixel/block-wise dependence to prevent sophisticated manipulations, such as VQ attacks. Three basic categories of pixel/block-wise dependence mechanisms have been described: deterministic (Type 1), non-deterministic (Type 2), or external information-based (Type 3). Apart from the localisation, other desirable features have been explored in fragile watermarking literature, such as reversible embedding and self-recovery. Table 2.1 summarises the methods reviewed in this chapter.

A limitation common to most fragile watermarking schemes is that the detectors cannot differentiate between images watermarked using the same key. Therefore, a different key has to be used to watermark every single image. In practice, however, keeping track of the key associated with every image may become challenging as the number of host images grows. Fridrich's scheme [20] provides a suitable solution to this problem, as the same key can be used to watermark several images, each one associated with a unique image index. Nevertheless, this method may become susceptible to brute force attacks when using a smaller block-size. Chapter 4 investigates this problem and presents a secure method with improved localisation capabilities.

An adversary can crop a host image to remove "inconvenient" portions of the image, thereby changing the dimensions of the image. However, existing fragile watermarking methods with self-recovery capabilities are typically dependent of the scanning order followed by the system. Therefore, not a single pixel can be restored when a host image has been cropped. Chapter 5 presents a method resilient to cropping with improved localisation and self-recovery capabilities, compared with the current state-of-the-art scheme in [65].

Table 2.1: Summary of the reviewed methods. Abbreviations used: PB/PV, public/private (authentication); LOC, localisation (capabilities); PW/BW, pixel-wise/block-wise; PBWD, type of pixel/block-wise dependence (recall the classification in Section 2.4); REV, reversible (embedding); REC, recovery. Methods marked by '*' are vulnerable to known attacks.

METHOD	PB/PV	LOC	PW/BW	PBWD	REV	REC
Wong [23]	PB/PV	yes	BW	*	no	no
Wong and Memon [24]	PB/PV	yes	BW	3	no	no
Ouda *et al.* [25]	PB	yes	BW	3	no	no
Suthaharan [26]	PV	yes	BW	3	no	no
Fridrich [20]	PB/PV	yes	BW	3	no	no
Celik *et al.* [27, 28]	PB/PV	yes	BW	1	no	no
Li and Si [29]	PV	yes	BW	1	no	no
He *et al.* [30]	PV	yes	BW	2	no	no
Yeung and Mintzer [36]	PV	yes	PW	*	no	no
Fridrich [39]	PV	yes	PW/BW	3	no	no

Continued on next page

Table 2.1 – continued from previous page

METHOD	PB/PV	LOC	PW/BW	PBWD	REV	REC
Lu *et al.* [40]	PV	yes	PW	2	no	no
Liu *et al.* [41]	PV	yes	PW	2	no	no
Zhang and Wang *et al.* [31]	PV	yes	PW	2	no	no
He *et al.* [42]	PV	yes	BW	2	no	no
Hosinger *et al.* [43]	PV	no	-	-	yes	no
Fridrich *et al.* (I) [44]	PV	no	-	-	yes	no
Fridrich *et al.* (II) [44]	PV	no	-	-	yes	no
Goljan *et al.* [45]	PV	no	-	-	yes	no
Celik *et al.* [46, 47]	PV	no	-	-	yes	no
Xuan *et al.* [50]	PV	no	-	-	yes	no
Xuan *et al.* [51]	PV	no	-	-	yes	no
Celik *et al.* [48, 49]	PB/PV	yes	BW	1	yes	no
Fridrich *et al.* [52]	PV	no	-	-	yes	no
Tian [53]	PV	no	-	-	yes	no
Alattar [54, 55]	PV	no	-	-	yes	no
Stach and Alattar [56]	PV	no	-	-	yes	no
Kamstra and Heijmans [57]	PV	no	-	-	yes	no
Kallel *et al.* [58]	PV	no	-	-	yes	no
Ni *et al.* [59, 60]	PV	no	-	-	yes	no
Hwang *et al.* [61]	PV	no	-	-	yes	no
Kim *et al.* [62]	PV	no	-	-	yes	no
Yeh *et al.* [63]	PV	no	-	-	yes	no
Lee *et al.* [10, 11]	PV	yes	BW	2	no	yes
Zhang and Wang [65]	PV	yes	PW/BW	*	no	yes
Zhang *et al.* [66]	PV	yes	PW/BW	3	no	yes

Continued on next page

Table 2.1 – continued from previous page

METHOD	PB/PV	LOC	PW/BW	PBWD	REV	REC
Zhang and Wang [67]	PV	yes	BW	3	yes	yes
Fridrich and Goljan [12]	PV	yes	BW	2	no	yes
Lin *et al.* [68]	PV	yes	BW	2*	no	yes
Wang and Chen [70]	PV	yes	BW	2*	no	yes
Lee and Lin [72]	PV	yes	BW	2*	no	yes
Zhang *et al.* [73]	PV	yes	PW/BW	3	no	yes

Chapter 3

Exposing Forgeries by Means of Image Forensics

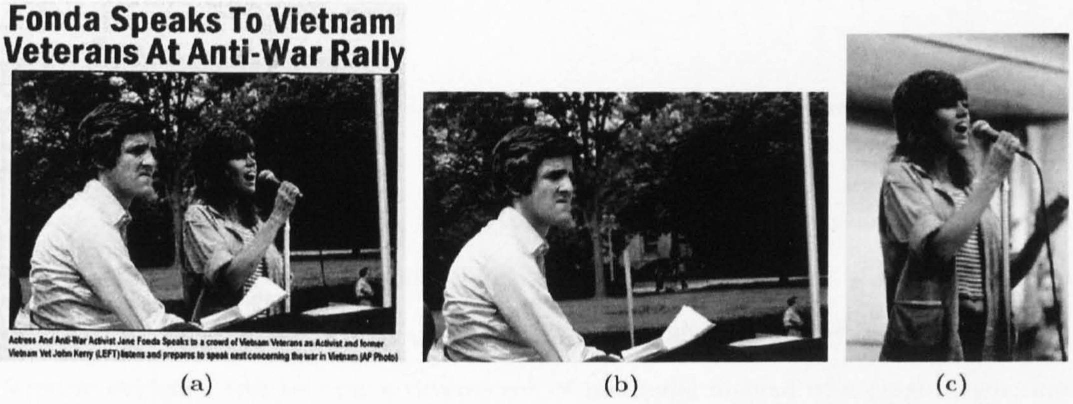
3.1 Introduction

This chapter focuses on forensic techniques intended to detect possible manipulations in digital images. Forensic methods that serve other purposes, such as ballistics or steganalysis[1], are beyond the scope of this investigation.

The use of digital re-touching tools in advertisements and photographs published in glossy magazines is generally accepted for aesthetic purposes – e.g. removing skin imperfections. However, some images can be doctored with the deliberate intention of adding credibility to a fabricated story. In fact, in some cases, such images circulate through the news media and other public outlets, which are expected to portray reality. For example, on the run up to the 2004 US presidential elections, the photograph[2] in Figure 3.1(a) was released to discredit the campaign of the candidate John Kerry, who appears to share a stage with the political activist Jane Fonda, in a rally against the Vietnam war in 1971. Later on, it was demonstrated that the image, falsely credited to

[1] Steganalysis is commonly included in the field of image forensics. Interested readers are referred to [2, 6] for a friendly introduction to steganography and steganalysis.

[2] The three images were obtained from the San Francisco Chronicle, February 20, 2004.



Fonda Speaks To Vietnam Veterans At Anti-War Rally

Actress And Anti-War Activist Jane Fonda Speaks to a crowd of Vietnam Veterans as Activist and former Vietnam Vet John Kerry (LEFT) listens and prepares to speak next concerning the war in Vietnam (AP Photo)

(a) (b) (c)

Figure 3.1: Example of a doctored image released during the 2004 US presidential election. (a) Image forgery (2004). (b) Photo of John Kerry (1971). (c) Photo of Jane Fonda (1972).

Associated Press, had been doctored by combining two photographs taken at different locations in different years [74]. The original photographs are shown in Figures 3.1(b) and 3.1(c). Interesting accounts of other historical photographic forgeries can be found in [74, 75].

In Chapter 2, it was shown that fragile watermarking can be used not only to detect and localise tampered regions in a host image, but also to restore the altered content. In practice, however, this approach is limited to controlled environments with specially equipped cameras, as the watermarks have to be inserted at the time of capturing [5, 76]. This situation has motivated the development of passive techniques to examine the content of images in a forensic fashion to identify possible traces of tampering, in the absence of authentication information generated *a priori*, such as watermarks or digital signatures.

This chapter aims to provide a concise, and yet complete, picture of image forensics for the purpose of tampering detection. Section 3.2 discusses some underlying principles generally accepted in this research field. The existing variety of image forensic tools is reviewed in Section 3.3 and a summary of the chapter is presented in Section 3.4.

3.2 Basic Principles

Image forensics is an incipient research field that has received an increasing amount of attention over the recent years[3]. Typically, these techniques work on the assumption that, even when it is difficult to tell by bare-eye examinations whether an image has been doctored, the distortions may alter the underlying statistics of the images. In practice, given the plethora of possible distortions an image may be subjected to, forensic evidence will rely on a diverse set of methods, instead of a single algorithm [5, 77].

Some concerns have been expressed as to whether the evidence derived from forensic techniques is reliable enough, as some countermeasures could be developed to deceive the forensic algorithms [77, 78]. However, the countermeasures may leave traces that could be detected by alternative forensic algorithms, leading to a cat-and-mouse race between forensic and counter-forensic techniques. It is widely believed that such emerging competition framework will contribute to a clearer and realistic picture of the field, and motivate the development of a more robust set of forensic algorithms. The desirable outcome is that the knowledge and computational resources required to develop and conduct effective attacks, capable of deceiving such algorithms, will increase making the generation of undetectable image forgeries more and more difficult, yet never impossible.

3.3 Literature Review

The diverse range of existing image forensic tools can be roughly divided into six groups, each of which will be separately addressed in this section. In an effort to provide a concise, yet complete, picture of this field, a representative set of the most effective methods will be reviewed in every group. Section 3.3.1 focuses on the problem of

[3] The research group led by Prof. Farid has made a valuable effort to keep an up-to-date database that includes existing and emerging methods on image forensics. The database is available at: `http://www.cs.dartmouth.edu/~farid/dfd/index.php/publications` (August, 2009)

detecting regions cloned within the same image. Methods aimed at detecting traces of interpolation or splicing are described in Sections 3.3.2 and 3.3.3, respectively. Forensic tools designed to detect inconsistencies in the artifacts introduced by lossy compression or the optical imaging systems used in digital cameras are reviewed in Sections 3.3.4 and 3.3.5, respectively. Finally, Section 3.3.6 describes techniques to identify inconsistencies in the lighting conditions of a doctored image.

3.3.1 Detection of duplicates

Perhaps one of the most common manipulations consists in concealing undesirable elements under a region of pixels copied from another part of the same image. In doing so, the replicated portion can be subjected to further distortions either to suit better the scene in the image or as an attempt to hamper the detection of the manipulation. Moreover, when designing a duplicate detector, additional post-processing operations, such as lossy compression, should be considered.

Fridrich *et al.* [5] demonstrated that the excessive computational cost of an exhaustive search would limit its use for only small images. Hence, they proposed a method, whereby the DCT of overlapping blocks of pixels is computed first. Then, the block coefficients are lexicographically sorted to reduce the computational cost of the search phase. Finally, duplicates are identified by grouping pairs of similar blocks separated by the identical spatial offsets. In [76], the pixel blocks are mapped to a dimension-reduced representation by means of the principal component analysis (PCA), thereby reducing the computational cost of the search while improving its robustness against JPEG compression. In Luo *et al.*'s method [79], the colour components of every block are used to generate feature vectors robust to JPEG compression and additive white Gaussian noise (AWGN). In [80], the pixel blocks are sorted using a k-dimensional tree to improve the efficiency of the search. Alternative approaches have been proposed to detect duplicates that have undergone blur degradation, contrast changes or sophisticated retouching filters [81, 82]. However, the methods in [5, 76, 79–82] are very sensitive to geometric changes in the duplicated areas.

This concern was first addressed by Myna *et al.* [83]. Here, overlapping blocks of wavelet coefficients in the coarser sub-band of the lower resolution level are mapped to log-polar coordinates. The resulting blocks are lexicographically sorted and analysed to identify similar pairs. The formed pairs are iteratively filtered by discarding those that do not fulfil the similarity criterion in the subsequent wavelet resolution levels. Duplicates affected by rotation and scaling can be detected by this scheme. In Lin *et al.*'s scheme [84], overlapping blocks of pixels are mapped to luminance-dependent features. Then, a radix sort-based algorithm is proposed to perform an efficient search for similar feature vectors. Nevertheless, this method cannot cope with scaling or rotation by arbitrary angles (it only supports 90°, 180° and 270°). Wang *et al.* [85] extended the method in [79] to use circle blocks and a Gaussian pyramid decomposition in order to detect rotated duplicates. Nevertheless the system is sensitive to other common geometric distortions, such as reflection or scaling. In Bayram *et al.*'s method [86], the pixel blocks are mapped to rotation-invariant 1-D descriptors derived from the Fourier-Mellin Transform (FMT). Then, a hash function is applied at every descriptor to identify duplicates very quickly. Nonetheless, such a rigid detection mechanism makes the system too sensitive to rotations of more than 10°. Huang *et al.* [87] presented a different approach, whereby rotation/scaling-invariant descriptors are extracted from the image by means of the Scale Invariant Feature Transform (SIFT). Thus, the algorithm is confined to the search for pairs of similar descriptors, which are subsequently refined to produce a final set of key points. Other SIFT-based methods have been proposed in [88, 89]. Compared with Huang's scheme, these schemes also estimate the parameters of the geometric transformation between the duplicated areas. Additionally, in [89], the estimated parameters are used to produce a transformed version of the image being analysed. Then, the areas of the duplicates are delimited by computing a correlation map between the two versions of the image. In Ryu *et al.*'s method [90], the Zernike moments of every pixel block are calculated to produce a feature vector invariant to rotation. The resulting vector are then lexicographically sorted to find potential duplicates. However, neither scaled nor reflected duplicates can be detected by this method.

3.3.2 Interpolation detection

To create a convincing composite photograph, it is often necessary to rotate, shrink or stretch some elements in the image. In doing so, the altered pixels are mapped onto a different sampling lattice, thereby altering the statistics of the manipulated pixels in a detectable way.

Popescu and Farid [91] investigated the correlations between the pixels in the image introduced by interpolation. The authors use the expectation/maximisation (EM) algorithm to estimate the probabilities of each pixel being correlated to its neighbours. These probabilities are collected in a probability map. Portions of the image that have been interpolated will produce some periodic patterns in the probability map, which can be identified as distinctive peaks in its frequency spectral representation obtained by means of the discrete Fourier transform (DFT). The authors proposed an exhaustive mechanism to detect automatically such peaks in order to determine whether or not the analysed image contains interpolated areas. The detection performance is consistently high for uncompressed images. This method has been simplified without significantly compromising its detection performance in [92]. Gallagher [93] proposed a method to detect images that have been digitally zoomed. To this end, the author studied the periodic patterns introduced by low-order interpolation (e.g. bilinear or bicubic interpolation) in the second derivative of the image. Then, a threshold-based algorithm was proposed to estimate the interpolation factor by identifying peaks in the frequency domain. However, neither rotation nor skewing is addressed in this method. Mahdian and Saic [94] provided an analytical framework to explain the periodic properties found in the covariance structure of interpolated images and their derivatives. The authors also proposed a detection mechanism, based on the Radon transform, that identifies periodic elements of the derivative of the image by detecting peaks above a predefined threshold in the frequency domain. The detection performance of the methods in [91–94] drops for images compressed with JPEG at quality factors below 95%. This is because the periodic patterns resulting from interpolation can be obliterated by the blocking artifacts introduced by JPEG compression. The methods described so far assume that the original image has never been compressed.

Some counter-forensic techniques have been proposed to hide the traces of interpolation, thereby defeating the aforementioned detection mechanisms [77, 78, 95]. Nonetheless, as mentioned in Section 3.2, these attacks may leave traces that could be detected by other forensic tools.

3.3.3 Splicing detection

The following methods are aimed at detecting photomontages generated by digitally splicing portions copied from two or more images without post-processing, such as edge blurring. This manipulation is commonly regarded as *splicing*.

The authors in [96] showed that the higher-order statistics of an image are usually perturbed by splicing[4]. Based on this observation, Ng and Sun [96, 98] use a trained support vector machine (SVM) to detect the presence of splicing. This works by identifying abrupt discontinuities in features derived from the normalised bispectrum (bicoherence) of the image, which measures the third-order correlation of three harmonically related Fourier frequencies in the image. In [99], the authors presented a detection mechanism that works on the assumption that splicing may perturb the correlation among pixel values in the image. The algorithm maps the analysed image to a feature vector comprised of a set of statistical moments of characteristic functions and elements of Markov transition probability matrix. A trained SVM finally examines the resulting feature vector to determine whether the image is either authentic or spliced. Qu *et al.*'s [100] method starts by identifying a set of "atypical" salient points

[4] A previous work, in [97], yielded similar conclusions when analysing composites of human speech signals.

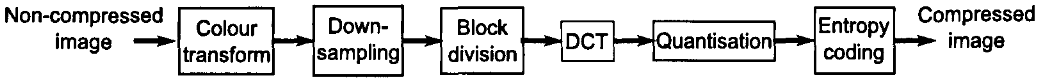


Figure 3.2: General framework of the JPEG compression algorithm.

in the image, some of which are expected to be located somewhere along the boundaries of the spliced region. A set of feature vectors, each one derived from the window of pixels around one of the salient points, are finally analysed with a trained hierarchical classifier.

3.3.4 Lossy compression-based approaches

All image lossy compression algorithms (e.g. JPEG or JPEG2000) embody some type of quantisation step to reduce the entropy of the information, thereby yielding higher compression rates. For example, given a RGB image, the standard JPEG compression algorithm follows the framework illustrated in Figure 3.2 [101]. First, the image is converted into the luminance/chrominance space (YCbCr). The space resolution of the Cb and Cr components is reduced (down-sampling) and then the three channels are split into blocks of 8×8 pixels. Every block is then transformed to the DCT domain, followed by the quantisation of every coefficient, which depends on the desired quality factor. Finally, the coefficients are arranged and encoded using a lossless compression algorithm (a variant of Huffman coding). The quantisation step introduces characteristic blocking artifacts which can serve as evidence that some sort of manipulation has taken pace.

Multiple JPEG quantisation

Consider a typical digital camera that stores the captured photographs in JPEG format. One of these images could be doctored and then re-saved using a different JPEG quality factor. The double JPEG compression introduces distinctive patterns in the histograms of DCT coefficients [102, 103] and alters the distribution of the first digits of the DCT blocks [104]. Based on these observations, various effective mechanisms have been proposed to detect the presence of double JPEG compression and estimate the primary quantisation table (i.e. the quantisation table used by the camera) [102–107]. It is important to note that the existence of double JPEG compression does not necessarily prove malicious manipulations, as the image could have been re-saved in JPEG format without the intention of changing the semantic meaning of the content. However, these forensic techniques could provide valuable insights into the history of an image, e.g. the authenticity of images presented as evidence in court could be denied if traces of double JPEG compression are found [108].

Some counter-forensics techniques have been proposed to compensate the distortion introduced by double JPEG compression by means of additive noise with an specific distribution [109, 110]. However, whether these attacks leave detectable traces is still an open problem.

Inconsistencies in JPEG blocking artifacts

To balance the trade-off between the size and the quality of the captured images, digital cameras produced by different manufacturers typically employ a distinctive JPEG quantisation table [111, 112]. For that reason, the introduced blocking artifacts also vary from camera to camera. Thus, a forgery image generated as a composite of two or more portions of JPEG compressed images will likely exhibit inconsistencies in the blocking artifacts, which can be be detected to expose the forgery. Another source of detectable inconsistencies in the blocking artifacts can be introduced by conventional retouching tools, re-sampling, filtering, etc.

In [113, 114] the authors noted that the typical shape of a DCT coefficient histogram of a tampered image can be viewed as the sum of two histograms with different distributions; one of the histograms describes the properties of the genuine blocks while the other describes the properties of the altered blocks. The authors accumulate the probabilities estimated using all the DCT coefficient histograms of every block to calculate a block posterior probability map (BPPM) by means of a Bayesian approach. The resulting BPPM is finally analysed with a trained SVM to decide whether the image has been tampered and, if so, produce a localisation map to identify the altered blocks. Luo *et al.* [115] characterise the JPEG blocking artifacts by calculating the difference among adjacent pixel values within each block and across the boundaries of neighbouring blocks; the differences are typically smaller within the blocks. A trained SVM is used to discriminate between the typical blocking artifacts introduced by a single JPEG compression and atypical artifacts resulting from cropping and re-compression. Nonetheless, the tampered regions are not localised by this method. Ye *et al.* [116] proposed a fast algorithm to calculate the JPEG quantisation table using the second derivative of the frequency spectrum of the DCT coefficient histograms. The estimated quantisation table is used to calculate a blocking artifact measure (BAM) for every DCT block in the image. Significant variations in the BAM are regarded as traces of tampering. Kirchner and Gloe [117] analysed the situation where the original image was JPEG compressed before the tampering took place. In this scenario, the forgery image can be deliberately re-compressed to generate a second set of JPEG periodic artifacts. Thus, portions of the forgery image that have been scaled will exhibit shifted versions of the JPEG peaks in the frequency spectrum of the probability maps, which can be readily identified using a tailored version of the detector in [92]. Nonetheless, the detection capabilities of the scheme are limited to scaling.

3.3.5 Camera-based approaches

An image captured with a conventional digital camera undergoes a series of processing stages, which unwittingly introduce a distinctive fingerprint into the image which can be examined in a forensic fashion to detect manipulations.

Camera response function

When a digital camera captures an image, the irradiance received by the charge-coupled device (CCD) sensor is mapped to a pixel value. The mapping, known as the camera response function (CRF), is commonly used to enhance the image, thereby introducing some non-linearity among adjacent pixels. Lin *et al.* [118] starts by describing the typical properties of the CRF from a collection of non-tampered images. The detection algorithm works on a set of patches of pixels manually selected along the edges of regions suspected to be part of a spliced portion. A trained SVM is then used to identify images that exhibit significant inconsistencies in the CRFs estimated from each patch. Nonetheless, the detection performance relies on the ability of the user to select a correct set of patches. In [119], the analysed image is manually segmented into three regions corresponding to the reliable region (assumed to be genuine), the suspicious region (spliced area) and a region containing the splicing boundary. A feature vector derived from the CRF estimated for the each region is analysed by a SVM that classifies authentic and spliced images. This method has been extended to overcome the dependency of the manual selection by incorporating an automated segmentation mechanism [120].

Demosaicking

Colour images are typically represented as a combination of three colour components, namely red, green and blue. Many commercial cameras are equipped with a single CCD and use a colour filter array (CFA) to capture colour. A CFA can be seen as a mosaic of colour filters (red, green and blue), with a specific arrangement, placed on a grid of photo-sensors. Each photo-sensor captures only one colour component for each pixel location. The two missing colour components of each pixel are estimated from its neighbouring samples, using a mechanism commonly referred to as *demosaicking* or CFA interpolation. This process introduces a periodical correlation among groups of pixels in the three colour channels. Popescu and Farid [91] investigated the effects of various demosaicking techniques and used an EM algorithm to estimate the probabilities of each pixel being correlated to its neighbours. The probabilities are used to form a probability map that is analysed in the Fourier domain to identify possible peaks –

a by-product of demosaicking. The absence of the traces of demosaicking in local regions of the image can be considered an evidence of manipulation. Gallagher and Chen [121] showed that demosaicking introduces a periodical pattern in the variance of pixels along a diagonal in a local neighbourhood, which can be efficiently detected in the frequency domain. The method can effectively identify regions that lack of such periodical patterns, which are deemed altered.

Some techniques have been proposed to introduce synthetic traces of demosaicking as an attempt to deceive the detection methods described above [122, 123]. Nonetheless, this will likely motivate the design of new detection tools to overcome these attacks.

Chromatic aberration

Another imperfection of the optical imaging systems that can be exploited for forensic purposes, which is termed *chromatic aberration*, occurs because of the inability of lens to focus light of all wavelengths (colours) to the same point on the sensor. Chromatic aberration manifests itself in two forms: longitudinal and lateral. Longitudinal chromatic aberration results in differences in the focal planes for different wavelengths. Lateral chromatic aberration (LCA) results in different wavelengths being projected to slightly different locations of the sensor; the separation increases as the distance from the optical axis increases. Johnson and Farid [124] estimated the parameters of the global LCA at the sub-pixel level using a brute-force mechanism. Then, the detection algorithm scans the image to identify significant inconsistencies in local regions, which are considered evidence of manipulation. To reduce the computational cost of this method, Gloe *et al.* [125] proposed to estimate the parameters of the global LCA using the displacement vectors calculated for different local regions in the image.

The main limitation of these methods is that they work on the assumption that the tampered region is sufficiently small not to interfere with the estimation of the global LCA. Moreover, more robust models of the LCA are necessary to consider the complex characteristics of the modern lens systems, in some cases combined with optical stabilisers, which may disturb the LCA in a manner that cannot be predicted with the existing models [125].

3.3.6 Inconsistencies on lighting conditions

It is often difficult to generate a photomontage, because the lighting conditions of objects or people taken from different photographs do not match perfectly. Johnson and Farid [126] worked on the algorithm proposed in [127], which allows them to estimate the direction of a point light source from a single image, i.e. without the knowledge of the 3-D surface. After manually selecting specific parts along the silhouette of the individual objects, the light source is estimated and compared to reveal possible inconsistencies. Reported results demonstrate the effectiveness of this approach in environments with a single (or dominant) directional light source, e.g. outdoors photographs on a clear sunny day. This model has been extended, in [128], for complex lighting environments including multiple light sources or diffuse lighting conditions. In [129], the authors use the bright spot light (known as specular highlight) in the eyes of people in the scene to estimate the direction of the light source. If the light source estimated for different people in the photograph is inconsistent, the image is deemed a fake.

3.4 Summary and Discussion

The emerging field of image forensics describes mechanisms capable of detecting traces of tampering in digital images, in the absence of watermarks of digital signatures. Typically, such mechanisms work on the assumption that the manipulations would alter the underlying statistics of the images, which can be detected to expose a counterfeit.

Due to the huge number of possible image distortions, forensic evidence will be collected from a diverse arsenal of methods, rather than from a single algorithm. The set of image forensic tools reviewed in this chapter is summarised below:

- Methods to detect duplicated regions within the same image: [5, 76, 79–90].

- Methods to detect traces of interpolation introduced by some geometric transformations commonly used when creating a composite of images: [91–94, 117]. Counter-forensic mechanisms to hide the traces of interpolation: [77, 78, 95].
- Methods to detect traces of splicing: [96, 98–100].
- Methods to identify inconsistencies in the artifacts introduced by lossy compression: [102–107, 113–116]. Counter-forensic techniques to compensate the distortion introduced by double JPEG compression: [109, 110].
- Methods to identify inconsistencies in the distortions introduced by optical imaging systems used in digital cameras: [91, 118–121, 124, 125]. Counter-forensic attacks to introduce synthetic traces of demosaicking: [122, 123].
- Methods to identify inconsistencies in lighting conditions: [126–129].

Image forensics is a nascent field with a promising potential to restore some of people's trust in digital images. The challenges ahead include the development of more robust detection mechanisms, as well as mechanisms to detect possible traces of manipulation introduced by some counter-forensic techniques proposed recently.

As pointed out, in Section 3.3.1, many schemes aimed at detecting duplicates within the same image are too sensitive to geometric changes in the replicated areas. Although some methods have been recently proposed to address this problem, none of them deal with reflection, rotation and scaling in a comprehensive manner. In Chapter 6, an automated detection method, that effectively identifies duplicated regions affected by this set of geometric transformations, is presented.

Chapter 4

Secure Fragile Watermarking Method with Improved Localisation

4.1 Introduction

In this chapter, the block-wise method proposed by Fridrich [20] is revisited as a suitable solution for the shortcomings resulting from protecting every single image, in a large data-set, using a different watermarking key. The security limitations of Fridrich's method against brute force attacks, especially in applications where higher localisation accuracy is required, are pointed out. Then, a method capable of enhancing the localisation accuracy, while hindering brute force attacks, is proposed.

Distinguishing among images obtained with the same capturing device is essential for security-related applications, such as surveillance cameras. To elaborate on this problem, consider two images that have been captured by the same hypothetical CCTV camera: Figure 4.1(a), captured at a time t_1, just a few seconds before a crime was committed, and Figure 4.1(b), captured at a different time t_2. The image in Figure 4.1(b) could be, either accidentally or maliciously, presented as evidence in trial. More-

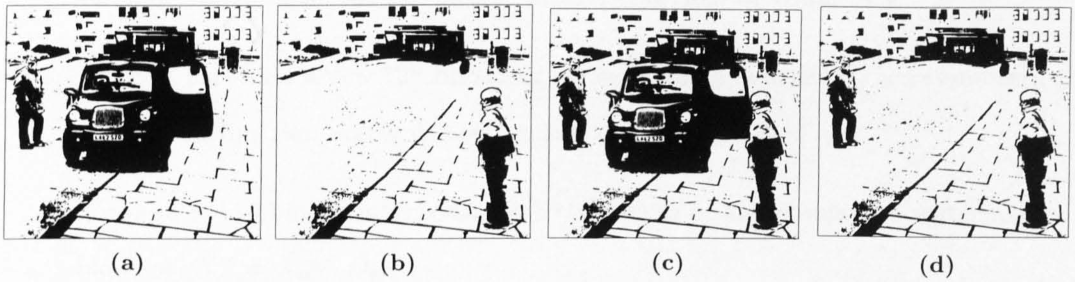


Figure 4.1: Images captured from an hypothetical CCTV camera and two possible counterfeits. (a) Image captured at time t_1. (b) Image captured at time t_2. (c) and (d) represent two possible counterfeits.

over, an adversary that have access to the two images could produce a counterfeit, such as those illustrated in Figures 4.1(c) and 4.1(d). Accepting any of these three images as valid evidence in court may lead to wrong conviction or acquittal.

Most of the fragile watermarking methods reviewed in Chapter 2, can cope effectively with this problem, as long as every image is watermarked using a different key. Nonetheless, keeping track of the secret key associated to every single image may become increasingly difficult as the number of captured images grows – i.e. a conventional CCTV camera produces over 2 million images every day when operating at 32 frames per second (fps). Fridrich's approach [20] provides an elegant solution to this problem, as the same key can be used to watermark various images, each one associated to a different image index. Nevertheless, in the present investigation, it has been found that this method can become susceptible to brute force attacks in applications wherein higher localisation is required. To elucidate on this problem, the steps followed by Fridrich's algorithm to watermark a block of 8 × 16 pixels, denoted as X_p, will be reproduced below.

1. Encode a 128-bit string d_p recalling the structure in Figure 2.2.
2. Compute a 128-bit string, $w_p = \mathcal{H}(\hat{X}_p) \oplus d_p$, where $\mathcal{H}(\cdot)$ is a cryptographic hash function, $\oplus$ denotes the XOR operator, $\hat{X}_p = 2\lfloor 2^{-1}X_p \rfloor$ and $\lfloor x \rfloor$ is the floor

function that maps x to the largest integer less than or equal to x.

3. Encrypt w_p to get a new 128-bit string, $e_p = \mathcal{E}(w_p, k)$, where $\mathcal{E}(\cdot)$ is a symmetrical encryption function and k is a secret key.

4. Form an 8×16 binary matrix D_p with the bits in e_p and compute a watermarked block as, $X_p^w = \hat{X}_p + D_p$.

Observe that, for an altered block, say $Z_a \neq X_p$, to go unnoticed, the following equations must be satisfied,

$$\begin{aligned} \mathcal{H}(\hat{X}_p) &= \mathcal{H}(\hat{Z}_a) \ , && (4.1) \\ X_p - \hat{X}_p &= Z_a - \hat{Z}_a \ , && (4.2) \end{aligned}$$

where $\hat{Z}_a = 2\lfloor 2^{-1} Z_a \rfloor$. Hence, the main challenge, for an attacker, is to generate a hash collision – i.e. a block Z_a that satisfies Equation (4.1). Observe, however, that this can be attempted blindly, without the knowledge of the key k. Once a hash collision has been found, the LSBs of Z_a can be readily replaced with the LSBs of X_p to fulfil Equation (4.2).

Finding a hash collision is known to be computationally infeasible because of the properties of cryptographic hash functions. Nevertheless, according to the well-known birthday paradox, brute force attacks could become viable when the block-size is sufficiently small [130]. For example, for a block-size of 8 × 8, roughly 2^{32} attempts would be necessary to find a couple of blocks that satisfy Equation (4.1), which is achievable with existing technology. The lack of flexibility of the authentication structure, in Figure 2.2, poses another limitation that renders this system unsuitable for applications where higher localisation accuracy is required. For example, it is not clear how the authentication structure should be rearranged to suit a block-size, say of 8 × 8.

These concerns are addressed in this chapter, which is structured as follows. A scheme is presented in Section 4.2, whereby higher localisation accuracy can be achieved without compromising the security against brute force attacks. The proposed scheme takes a different approach to verification, where blocks are collectively authenticated, after identifying significant concurrences in the messages extracted from every single block. Section 4.3 elaborates on the distortion and the security of the scheme, with an emphasis on how brute force attacks are effectively thwarted. Additionally, a reliable protocol is presented to resolve possible doubts about the source and/or the integrity of a host image. In the experimental results reported in Section 4.4, the proposed method is compared with five existing methods, in terms of localisation performance in presence of a various types of attacks. Finally, the chapter is summarised and concluded in Section 4.5.

4.2 Proposed Scheme

To enhance the localisation accuracy, while hindering brute force attacks, the hash function is computed over the secret key together with the pixel block. Additionally, a different approach to verification is presented to eliminate the need for duplicated data in the structure embedded in each block. Instead, the authenticator goes through the bit strings extracted from all the blocks to identify significant concurrences, which are subsequently employed to authenticate every single block. As a result, Fridrich's scheme has been enhanced to provide:

- *Automated detection and localisation* – Any manipulation can be detected and properly localised in a single automatic verification stage.

- *Higher security* – The structure embedded in each block, illustrated in Figure 4.2, involves no duplicated data (compare with Figure 2.2). Section 4.3.2 discusses how this increases the security of the scheme.

- *Logo-based authentication* – The pixels of a binary logo are spread over the LSB of the embedded authentication structure. Although, strictly speaking, this is

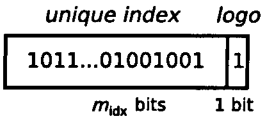


Figure 4.2: Watermarking structure encoded for each block in the proposed scheme.

not an essential requirement for authentication, "meaningful" watermarks – e.g. logos – can be beneficial for non-technical users [131]. For example, in a judicial enquiry, an extracted logo may help to convince the jury about the original source of an image.

- *Flexible localisation accuracy and distortion boundaries* – the localisation accuracy requirements, as well as the allowed embedding distortion, may vary from application to application. Thus, watermarking schemes with such an adaptability can be suitable for more applications. Both the block-size and the length of the authentication message can be easily adjusted, at the expense of increasing/decreasing the security of the proposed scheme.

The general framework will be introduced, before proceeding with the details of the proposed scheme. To thwart VQ attacks, every single block is watermarked with a unique index along with a bit of a user-defined binary logo by employing a secret key. A possible application scenario is depicted in Figure 4.3 (the watermark embedding system can be hard-wired within the camera). The unique index is comprised of the serial number of the camera and an image index. Having received the secret key from a secure channel, the verification system automatically localises possible distortions in the extracted logo and discloses the retrieved image index. The unique index is not required at the verification process, as it is assumed that the user does not necessarily keep track of the image index associated to every single image[1]. Furthermore, the same secret key can be employed to authenticate various images, provided that a distinct image index is associated to every single image. In fact, the maximum number of images watermarked

[1] Self-descriptive image indices, such as timestamps, could be obviously advantageous.

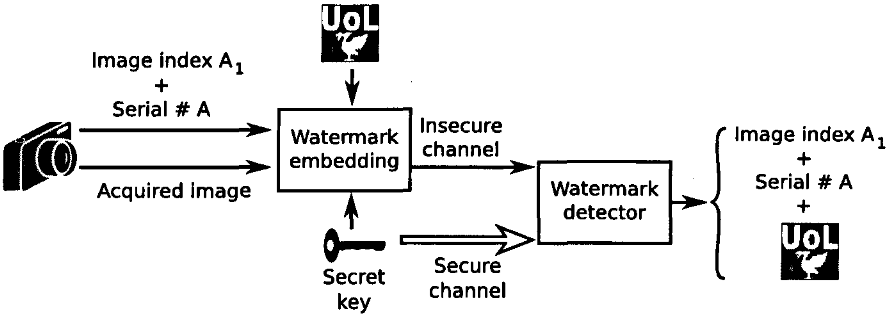


Figure 4.3: Practical scenario for the proposed scheme. A unique index, associated to each host image, is comprised of the serial number of the camera and an image index.

with the same key depends on the number of bits allocated for the image index in the authentication structure (Figure 4.2).

4.2.1 Embedding scheme

Consider an $n_1 \times n_2$ grey-scale image X, which is divided into non-overlapping blocks of $m_1 \times m_2$ pixels. Let X_p denote the p-th block in the image, for $p = 1, \ldots, (n_1 n_2 / m_{\text{bw}})$, where $m_{\text{bw}} = m_1 m_2$ is the total number of pixels in each block.

Consider a binary logo L of size $(n_1/m_1) \times (n_2/m_2)$, where $L_p \in \{0, 1\}$ denotes the p-th element in the logo. Let φ_X be a serial number and ϑ_X be an image index. The bits in φ_X and ϑ_X are concatenated to form the unique index, whose length, in bits, will be denoted by μ_X (it is assumed that $\mu_X < m_{\text{bw}}$).

The steps below are repeated to watermark every single block X_p:

1. Encode a binary structure as,

$$w_p = \varphi_X \,||\, \vartheta_X \,||\, L_p \ , \tag{4.3}$$

   where $||$ denotes concatenation of bits.

2. Compute a bit string, of length $(\mu_X + 1)$, as,

$$h_p = \mathcal{H}(k, \hat{X}_p, p) \ , \tag{4.4}$$

where $\hat{X}_p = 2\lfloor 2^{-1} X_p \rfloor$. The secret key k is a bit string whose length can be adjusted to satisfy the security requirements of the application.

3. Compute, $d_p = h_p \oplus w_p$.

4. Spread the bits in d_p over the LSB of the first $(\mu_X + 1)$ pixels in X_p.

All the watermarked blocks are assembled together to form a watermarked image X^w. Observe that the structures encoded for any arbitrary pair of blocks, say w_a and $w_{a'}$, will only differ in their LSB. This situation is exploited at the verification stage. Furthermore, note that the secret key k is an input of the hash function in Equation (4.4). Even though, a similar idea has been explored in [24], Section 4.3.3 elaborates on how this mechanism can hinder brute force attacks.

4.2.2 Extraction scheme

Let Y be an input $n_1' \times n_2'$ image, which is divided in non-overlapped blocks of $m_1 \times m_2$ pixels, where Y_p denotes the p-th block, for $p = 1, \ldots, m_Y$, where $m_Y = (n_1' n_2' / m_{\text{bw}})$ is the total number of blocks in Y. The following steps describe how to retrieve a bit string w_p' from every single block Y_p.

1. Encode a bit string d_p' by concatenating the LSB of the first $(\mu_X + 1)$ pixels in Y_p.

2. Compute the bit string, of length $(\mu_X + 1)$, as,

$$h_p' = \mathcal{H}(k', \hat{Y}_p, p) \ , \tag{4.5}$$

where $\hat{Y}_p = 2\lfloor 2^{-1} Y_p \rfloor$ and k' is the key provided to the extraction process.

3. Retrieve the authentication bit string as, $w'_p = h'_p \oplus d'_p$.

Once an authentication bit string has been retrieved from every single block, proceed with the following verification scheme.

4.2.3 Verification scheme

Let $\mathbf{B}$ be the set that contains all the blocks in Y, i.e.,

$$\mathbf{B} = \{Y_1, \ldots, Y_{m_Y}\} \ .$$

Split $\mathbf{B}$ into disjoint subsets $\mathbf{B}_1, \mathbf{B}_2, \ldots$, such that the same bit string $\breve{w}_1$ is retrieved from all the blocks in $\mathbf{B}_1$, $\breve{w}_2$ from the blocks in $\mathbf{B}_2$ and so forth.

Consider the case where Y is a watermarked, possibly altered, image – i.e. $Y \simeq X^w$ – and the correct key is provided to the extraction process. In this scenario, two authentication bit strings, say $\breve{w}_a$ and $\breve{w}_{a'}$, are expected to be identical except for their LSB. Furthermore, the cardinality of the sets $\mathbf{B}_a$ and $\mathbf{B}_{a'}$ is expected to be significantly greater than the rest of the subsets, i.e. $\forall_{i \neq a,a'} \ |\mathbf{B}_a \cup \mathbf{B}_{a'}| >> |\mathbf{B}_i|$. Without loss of generality, it will be assume that $\breve{w}_a$ is the bit string whose LSB is nought. Thus, if $|\mathbf{B}_a \cup \mathbf{B}_{a'}| > \tau_{\text{bw}}$, where $\tau_{\text{bw}} > 1$ is a predefined threshold, the intensity value of the p-th element of a bitmap is encoded as,

$$L'_p = \begin{cases} 0 & \text{if } w'_p = \breve{w}_a \\ 255 & \text{if } w'_p = \breve{w}_{a'} \\ 128 & \text{otherwise} \end{cases} \ . \tag{4.6}$$

Note that tampered blocks are encoded with a different intensity value (128) to ease user inspection. Additionally, the μ_X MSBs of $\breve{w}_a$, identical to the μ_X MSBs of $\breve{w}_{a'}$, are decomposed into the serial number and the image index to report them to the user.

If the block-size is large enough (see Section 4.3.2), the cardinality of all the subsets in a non-watermarked, or thoroughly tampered, image is expected to be less than the predefined threshold, i.e. $\forall_i \, |\mathbf{B}_i| < \tau_{\text{bw}}$. In this case, the detector deems the image non-watermarked with the key k' (a wrong key could have been used).

An especial scenario occurs when more than one pair of authentication bit strings, say $\breve{w}_a$, $\breve{w}_{a'}$, and, $\breve{w}_b$, $\breve{w}_{b'}$, are pair-wise identical except for their LSB. Additionally, the cardinality of the union of each pair of sets is greater than the threshold, i.e. $|\mathbf{B}_a \cup \mathbf{B}_{a'}| > \tau_{\text{bw}}$ and $|\mathbf{B}_b \cup \mathbf{B}_{b'}| > \tau_{\text{bw}}$. Under these circumstances, a different bitmap can be encoded for each pair of authentication bit strings by employing Equation (4.6).

Observe that the method thwarts VQ attacks, as the serial number φ_X, the image index ϑ_X and the block index p, together deter the creation of VQ code-books. Furthermore, the system is capable of providing one or more bitmaps for user inspection, as well as a report including the integrity of the host image, the retrieved serial number and the image index. All these features can be obtained even without the image index of the received image.

4.3 Analysis

The distortion induced by the embedding method is analysed in this section, along with the security of the whole scheme. Additionally, a reliable protocol is proposed to resolve possible disputes about the source or integrity of a host image.

4.3.1 Distortion

The worst case scenario will be analysed to assess the distortion inflicted on host images. In such scenario, the LSB of the $(\mu_X + 1)$ pixels in each block are altered by the embedding process. In such a situation, the MSE of a watermarked image, X^w, with respect to the original one, X, would be,

$$\text{MSE} = \frac{1}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} (X(i,j) - X^w(i,j))^2 = \frac{\mu_X + 1}{m_1 m_2} \ . \tag{4.7}$$

Hence, the PSNR of a host image will always be,

$$\mathrm{PSNR} \geq 10\log_{10}\left(\frac{m_1 m_2 \max_X^2}{\mu_X + 1}\right) , \tag{4.8}$$

where $\max_X$, typically set to be 255, is the maximum possible value in X.

4.3.2 Security

For the following analysis, it will be assumed that the output of $\mathcal{H}(\cdot)$ is drawn from a uniform distribution. This is, indeed, a well-known property of standard cryptographic hash functions, e.g. the standard algorithms MD5 [132] or SHA [133].

The probability that a manipulation will go unnoticed by the detector will be determined first. Let E_1 be the event that the μ_X MSBs of the bit string extracted from a tampered block, Z_{p_1}, is identical to the μ_X MSBs of the bit string originally embedded into the genuine block $X_{p_1}^w$, provided that $Z_{p_1} \neq X_{p_1}^w$. The probability that the event E_1 occurs is $\mathcal{P}_{E_1} = 2^{-\mu_X}$. Let $\mathcal{X}_1$ be a random variable that indicates the number of occurrences of E_1. Thus, the probability that η altered blocks will be mistaken for genuine blocks can be defined as,

$$\mathcal{P}_{\mathcal{X}_1}(\mathcal{X}_1 = \eta) = \prod_{i=1}^{\eta} \mathcal{P}_{E_1} = \prod_{i=1}^{\eta} \frac{1}{2^{\mu_X}} = \frac{1}{2^{\eta\mu_X}} . \tag{4.9}$$

To determine the likelihood that a non-watermarked image will be deemed genuine, consider a bit string, say $\breve{w}_a$, extracted from an arbitrary block, say Y_{p_1}, in a non-watermarked image. Let E_2 denote the event that the μ_X MSBs of the bit string extracted from another block, say Y_{p_2}, is identical to the μ_X MSBs of $\breve{w}_a$, provided that $p_1 \neq p_2$. In accordance with the well-known birthday paradox [134], the probability that E_2 will occur is given by,

$$\mathcal{P}_{E_2}(2^{\mu_X}, m_Y) = 1 - \frac{2^{\mu_X}}{(2^{\mu_X} - m_Y)!\, 2^{\mu_X\, m_Y}} . \tag{4.10}$$

Let E_3 be the event that the μ_X MSBs of the bit string retrieved from another block matches the μ_X MSBs of $\breve{w}_a$. The probability that E_3 will occur is $\mathcal{P}_{E_3} = 2^{-\mu_X}$.

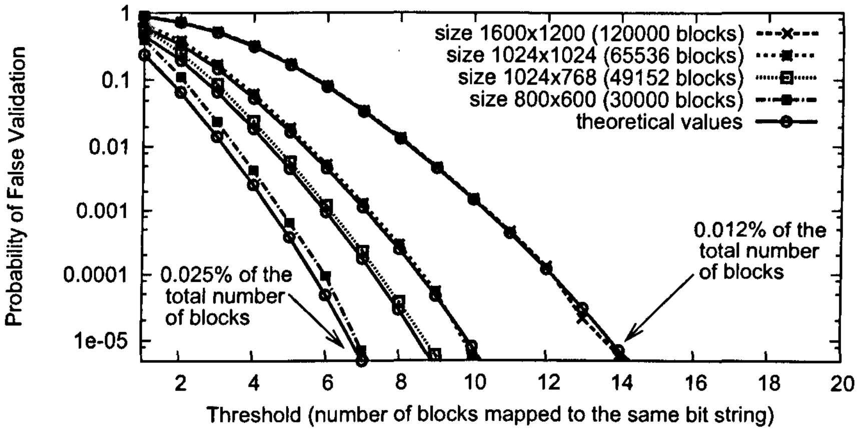


Figure 4.4: Influence of the total number of blocks over the definition of the threshold. false positive probability empirically computed in 956 images of different sizes (dashed lines); solid lines represent the theoretical values.

Let $\mathcal{X}_3$ be a random variable that indicates the number of occurrences of the event E_3. Recall that a non-watermarked image will be regarded as genuine only if the event E_3 occurs at least $\mathcal{T}_{\text{bw}}$ times. Therefore, the probability that the E_3 will occur, at least, $\mathcal{T}_{\text{bw}}$ times can be modelled as the remainder of a cumulative binomial distribution [135],

$$\mathcal{P}_{\mathcal{X}_3}(\mathcal{X}_3 \geq \mathcal{T}_{\text{bw}}) = 1 - \sum_{i=0}^{\mathcal{T}_{\text{bw}}-1} \begin{pmatrix} m_Y \\ i \end{pmatrix} \mathcal{P}_{E_3}^{i}(1 - \mathcal{P}_{E_3})^{m_Y - i} , \tag{4.11}$$

where,

$$\begin{pmatrix} m_Y \\ i \end{pmatrix} = \frac{m_Y}{i!\,(m_Y - i)!} ,$$

is the binomial coefficient – i.e. the possible ordered sequences of i blocks out of m_Y. Thus, the probability that a non-watermarked image will be regarded as genuine is,

$$\mathcal{P}_{\text{fa}} = \mathcal{P}_{E_2}\mathcal{P}_{\mathcal{X}_3} . \tag{4.12}$$

From Equation (4.11), it is clear that the definition of the authentication threshold, τ_{bw}, is important to keep a low probability of falsely validating a non-watermarked image. However, this probability increases as the number of blocks raises. To better illustrate this fact, the detector was set up to use blocks of 4 × 4 pixels[2]. Then, a data set of 956 non-watermarked images were submitted for authentication. The data set included images of size: 1600 × 1200, 1024 × 1024, 1024 × 768 and 800 × 600. Figure 4.4 shows the probability of falsely validating a non-watermarked image empirically computed for each image-size (dashed lines), compared with the theoretical values estimated by Equation (4.12) (solid lines). Observe that, for an image sized 800 × 600, the probability of false validation would be roughly 4×10^{-5} if the threshold were set to be 7 (approximately 0.025% of the total number of blocks). To achieve a comparable probability of false validation of an image sized 1600 × 1200, the threshold should be set to be 14 (approximately 0.012% of the total number of blocks). In the experiments presented in Section 4.4, the probability of false validation was estimated to be virtually zero, as the threshold was set to be 0.1% of the total number of blocks (this can be calculated using Equation (4.12)).

4.3.3 Hindering brute force attacks

To demonstrate that finding a hash collision is insufficient to perform a successful attack, consider an arbitrary watermarked block, say X_a^w (recall the embedding scheme in Section 4.2.1). For an altered block, say $Z_b \neq X_a^w$, to go undetected, the following equations must be satisfied,

$$h_a = \mathcal{H}(k, \hat{Z}_b, b) \ , \tag{4.13}$$

$$X_a^w - \hat{X}_a^w = Z_b - \hat{Z}_b \ . \tag{4.14}$$

[2] Even though this block-size would be too insecure for any practical application, it was chosen for mere illustrative purposes.

where $h_a = \mathcal{H}(k, \hat{X}_a^w, a)$. We will disregard the restriction stated in Equation (4.14), as it can be readily circumvented. Furthermore, it is assumed that the secret key k is unknown for attackers. This is a reasonable assumption, as, otherwise, k could be employed to watermark any image without the consent of the legitimate watermark owner. Since h_a and a can be easily determined from a host image, a brute force approach could be attempted to find both a key k' and a block Z_b such that,

$$h_a = \mathcal{H}(k', \hat{Z}_b, b) \ . \tag{4.15}$$

To fulfil Equation (4.13) though, it is necessary that,

$$\mathcal{H}(k, \hat{Z}_b, b) = \mathcal{H}(k', \hat{Z}_b, b) \ . \tag{4.16}$$

Nevertheless, because of the properties of cryptographic functions, whenever $k' \neq k$, the probability that Equation (4.16) holds is $2^{-\mu_X}$. This is equivalent to the probability that an arbitrary distortion will go undetected (recall Section 4.3.2).

Evidently, the security of the proposed scheme relies on the strength and secrecy of the key k. The only option would be to exhaustively search through the key space to discover k. However, such an alternative would be utterly unfeasible if the length of k – i.e. the key space – is chosen to be sufficiently large. This is because, by analogy with symmetric cryptographic systems, the number of attempts necessary to find the correct key by brute force attacks is expected to be half of the size of the key space. For example, if k is a string of 1024 bits, it is expected that, on average, an exhaustive search would find the key after 2^{1023} trials [130].

4.3.4 A reliable and flexible protocol

The reliability of private authentication methods can be questioned because of some flaws in the protocol. For example, consider the case where an image is presented in court as evidence to implicate a person in a crime. The defence could claim that the

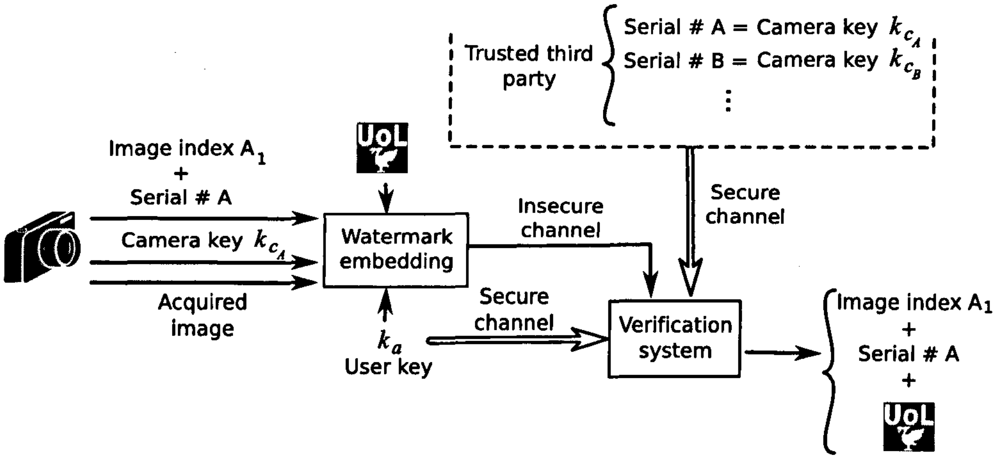


Figure 4.5: Proposed protocol for reliable authentication.

owner of the camera deliberately manipulated the image to embroil the suspect and then embedded the fragile watermark to make it to appear genuine. Such an allegation would be legitimate if it is considered that the owner of the camera is the only person who knows the secret key. This problem has been addressed for fingerprinting [136].

In the protocol proposed in [20], the secret key is physically wired in each camera, together with the watermark embedding system. Moreover, the serial number of each camera, and its corresponding secret key, are kept in a verification centre aimed to resolve any doubts about the origin and/or the integrity of an image. This protocol turns out to be impractical when the block-size is reduced, as the number of possible distinct image indices can be small.

To address this concern, the protocol depicted in Figure 4.5 is proposed. The serial number of each camera, and its associated secret key, are kept in a trusted third party. Such a key is physically wired at the camera and will never change. Additionally, the owner of the camera chooses a secret key, which can be conveniently updated from time to time. To deploy this protocol, Equations (4.4) and (4.5) are replaced with,

$$h_p = \mathcal{H}(k_c, k_a, \hat{X}_p^w, p) \ , \tag{4.17}$$

and,

$$h_p' = \mathcal{H}(k_c, k_a, \hat{X}_p^w, p) \ , \tag{4.18}$$

respectively, where k_c is the key associated with the camera and k_a is the user's key.

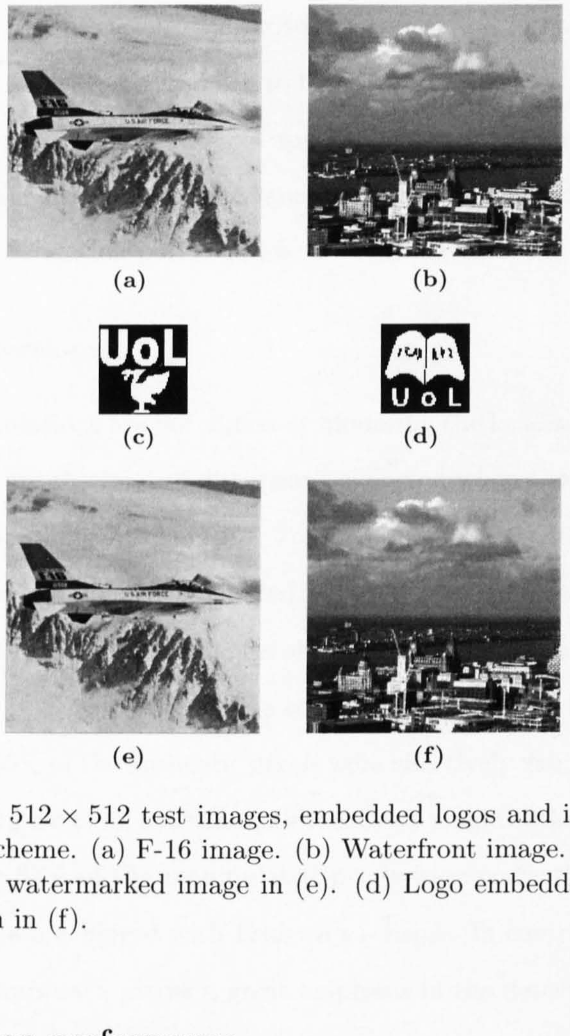
4.4 Experimental Results

In the following experiments, the system was set to use a block-size of 8 × 8, using authentication structures of 64 bits. The bit string retrieved from the standard SHA-256 hash function was trimmed off to 64 bits and the authentication threshold was set to $\tau_{\mathrm{bw}} = 3$; roughly 0.1% of the total number of blocks in the 512 × 512 test images shown in Figure 4.6(a) and 4.6(b). The same secret key, and distinct image indices, were used to watermark the test images with the logos depicted in Figures 4.6(c) and 4.6(d). The PSNR measures between the watermarked images, in Figures 4.6(e) and 4.6(f), and the original images were greater than 51 dB in both cases.

To assess the localisation performance, both the true positive rate (TPR) and the true negative rate (TNR), will be calculated,

$$\mathrm{TPR} = \frac{\mathrm{TP}}{P}, \text{ and, } \mathrm{TNR} = \frac{\mathrm{TN}}{N} \ , \tag{4.19}$$

where TP and TN are the number of pixels correctly deemed altered and genuine (by the detector), respectively, while P and N are the actual number of altered and genuine pixels (in the ground truth), respectively. For the sake of clarity, the TPR will be referred to as *detection performance* and the TNR as *validation performance.* Observe that the ideal localisation would include both a perfect detection performance (TPR=1) and a perfect validation performance (TNR=1).


(a) (b)

(c) (d)

(e) (f)

Figure 4.6: Original 512 × 512 test images, embedded logos and images watermarked with the proposed scheme. (a) F-16 image. (b) Waterfront image. (c) Logo embedded in (a) to obtain the watermarked image in (e). (d) Logo embedded in (b) to get the watermarked version in (f).

4.4.1 Localisation performance

The following experiments were conducted on various schemes reviewed in Section 2.5. For the sake of brevity however, only the results of the schemes that showed a better performance in each block-wise dependence type will be presented (recall the classification presented in Section 2.4): Li and Si's [29], He *et al.*'s [30] and Fridrich's [20] methods (Types 1, 2 and 3, respectively). To illustrate the limitations of these schemes when the same key is used to authenticate various images, both test images were watermarked using identical keys. The set-up suggested by the authors of each method was adopted. It is important to mention that tampered regions detected by Li an Si's method are originally encoded as noise-like areas in the output bitmap. This makes it difficult to conduct a fair comparison, as the validation performance achieved by this method would always be assessed to be around 0.5. To overcome this situation, a dilation (morphological operation) was applied on every bitmap encoded by Li and Si's method. Thus, the detection performance increased significantly, at the expense of slightly reducing its validation performance.

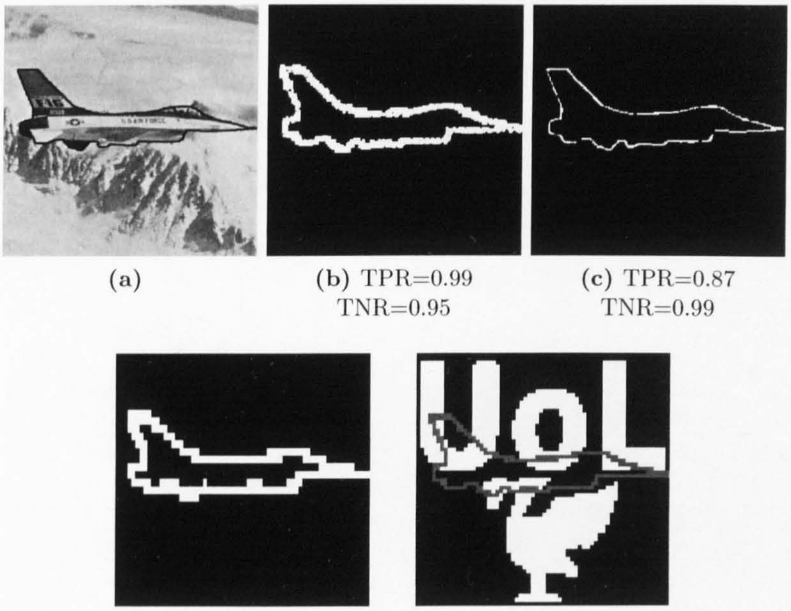
Conventional distortions

The following manipulations are not aimed at hindering the localisation performance of the detector. These are the kind of distortions expected when the attacker is unaware of the fragile watermarking mechanism. To exemplify a conventional distortion, the contour of the jet was manually delineated by employing the image editing software Gimp, as shown in Figure 4.7(a). Results obtained with Li and Si's system, after the dilation[3], are shown in Figure 4.7(b). The scheme achieved almost a perfect detection performance, while 95% of the authentic pixels were effectively validated. As illustrated in Figure 4.7(c), using He *et al.*'s method, almost every single authentic pixel was properly validated, while 87% of the manipulated pixels were correctly identified. Figure 4.7(d) shows the results achieved with Fridrich's scheme. In contrast to the preceding methods, Fridrich's approach places a great emphasis in the detection performance: it managed to identify all the manipulated pixels and validate 92% of the genuine pixels. In Figure 4.7(e) however, observe that the proposed method achieved a more accurate localisation, as the percentage of authentic pixels correctly validated increased by 5%, in comparison with Fridrich's results, without compromising the detection performance.

Inner-copy-and-paste attack

The background in the F-16 image was replaced with periodical duplicates of the upper-left block of pixels, as depicted in Figure 4.8(a). Figure 4.8(b) shows the results obtained with Li and Si's method: nearly all of the altered pixels were effectively identified,

[3] Results without the dilation: TRP=0.48 and TNR=0.97.



(a)

(b) TPR=0.99
TNR=0.95

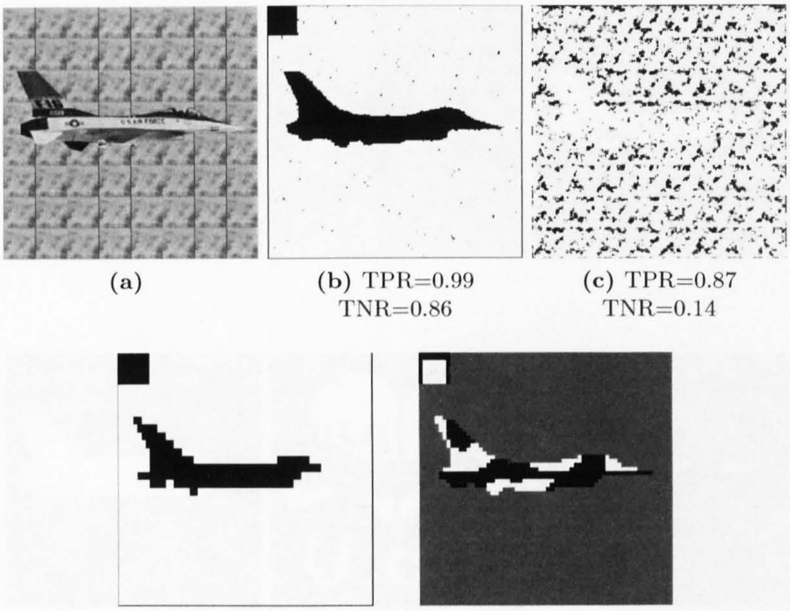
(c) TPR=0.87
TNR=0.99

(d) TPR=1.0
TNR=0.92

(e) TPR=1.0
TNR=0.97

Figure 4.7: Conventional distortion. (a) Image manipulated with a conventional image editing software. Authentication bitmaps retrieved with (b) Li and Si's, (c) He *et al.*'s, (d) Fridrich's methods, and (e) the proposed scheme.

while 86% of the authentic pixels were correctly validated. As shown in Figure 4.8(c), although He *et al.*'s scheme managed to identify virtually 9 out of every 10 of the tampered pixels, the validation performance is very poor. This problem, however, arises not because of the type of attack, but as a consequence of the large portion of tampered pixels. Results for Fridrich's and the proposed method are presented in Figures 4.8(d) and 4.8(e), respectively (the black and white regions in Figure 4.8(e) correspond to successfully retrieved portions of the embedded logo). Both schemes achieved a perfect detection performance, but the proposed method outperformed Fridrich's scheme by 0.1 in terms of validation performance.



(a)

(b) TPR=0.99
TNR=0.86

(c) TPR=0.87
TNR=0.14

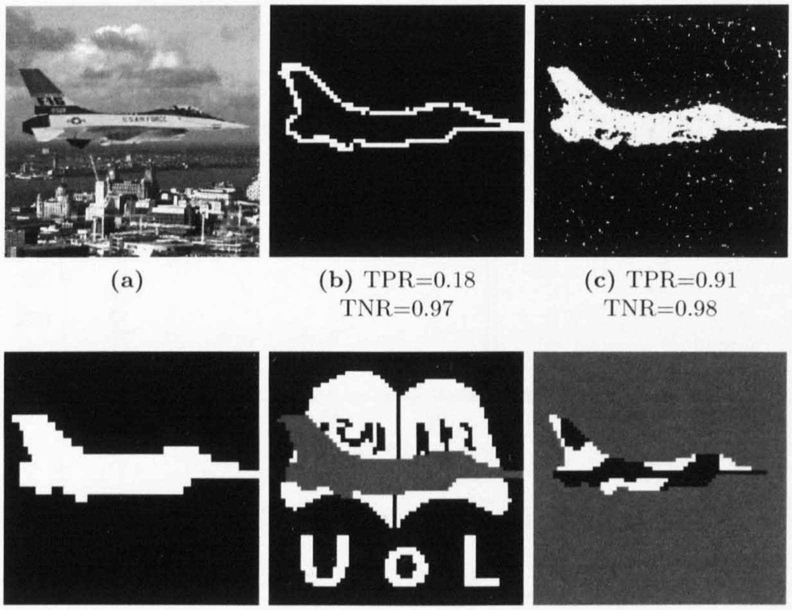
(d) TPR=1.0
TNR=0.75

(e) TPR=1.0
TNR=0.85

Figure 4.8: Inner-copy-and-paste attack. (a) Counterfeit image. Authentication bitmaps retrieved with (b) Li and Si's, (c) He *et al.*'s, (d) Fridrich's methods, and (e) the proposed scheme.

Outer-copy-and-paste attack

The F-16 jet was copied to the same location in the Waterfront image to form a counterfeit as in Figure 4.9(a). Results obtained with Li and Si's method are presented in Figure 4.9(b). Since the same key has been used to watermark the two images, the verification system is capable of only identifying the distortions that occurred all over the contour of the jet. Indeed, observe that the authentication bitmaps in Figures 4.7(b) and 4.9(b) exhibit remarkable similarities, even though the corrupted regions in Figures 4.7(a) and 4.9(a) are greatly distinct. Figure 4.9(c) shows that He *et al.*'s method managed to detect 91% of the tampered pixels and validate 98% of the genuine pixels. The automatic verification mechanism in Fridrich's method detected the distortions all over the contour of the jet, as in Li and Si's method. However, after scrutinising the bit strings extracted from every single block, it was possible to encode the bitmap in



(a) (b) TPR=0.18 TNR=0.97 (c) TPR=0.91 TNR=0.98

(d) TPR=1.0 TNR=0.96 (e) TPR=1.0 TNR = 0.98 (f)

Figure 4.9: Outer-copy-and-paste attack. (a) Counterfeit image. Authentication bitmaps retrieved with (b) Li and Si's, (c) He *et al.*'s, (d) Fridrich's methods. (e) and (f) bitmaps retrieved with the proposed scheme.

Figure 4.9(d). The proposed method automatically retrieved the two bitmaps in Figure 4.9(e) and 4.9(f), together with their respective image index. Results show that the proposed system managed to validate a slightly higher proportion of genuine pixels, compared with Fridrich's method.

4.4.2 Embedding distortion

Using 900 images, sized 480 × 640, from the Caltech-256 data-set [137], as test images, the proposed method was compared, in terms of embedding distortion, with the following schemes:

- Celik *et al.* [28] (Type 1),
- Li and Si [29] (Type 1).

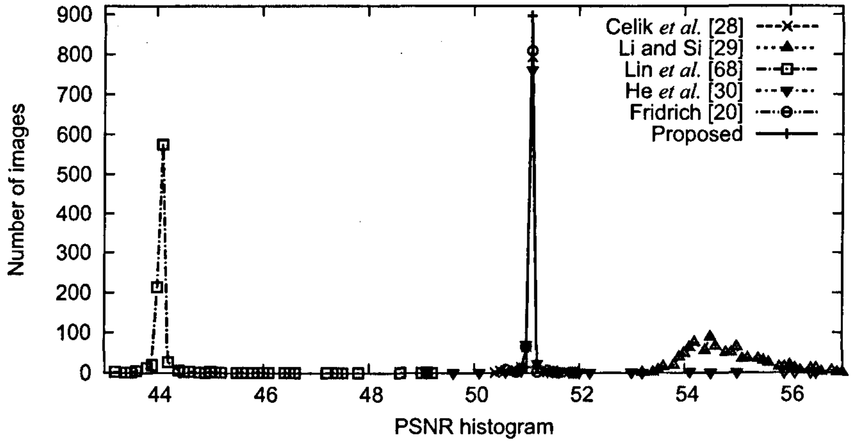


Figure 4.10: Comparison of embedding distortion.

- Lin *et al.* [68] (Type 2).
- He *et al.* [30] (Type 2).
- Fridrich's [20] (Type 3).

Every test image was watermarked with each method and its PSNR was computed. A histogram with the resulting PSNR values is shown in Figure 4.10. As expected, Lin *et al.*'s method induced a higher distortion, as it allocates the watermark in the 2 LSBPs of the host image; the average PSNR was assessed to be 44.2 dB. The embedding distortion induced by Celik *et al.*'s, He *et al*'s, Fridrich's and the proposed method was comparable. In these cases, the average PSNR was assessed to be 51.1 dB. Li and Si's scheme introduced a significantly lower embedding distortion, with over 90% of the PSNR values above 54 dB (the average PSNR was assessed to be 54.8 dB). This is because a portion of the wavelet coefficients is not altered by this watermarking system (recall Section 2.5.1).

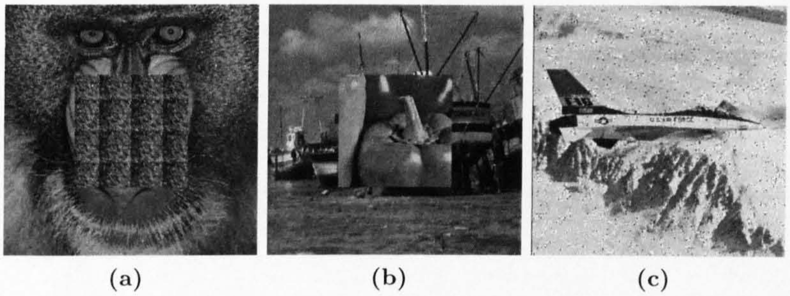

(a) (b) (c)

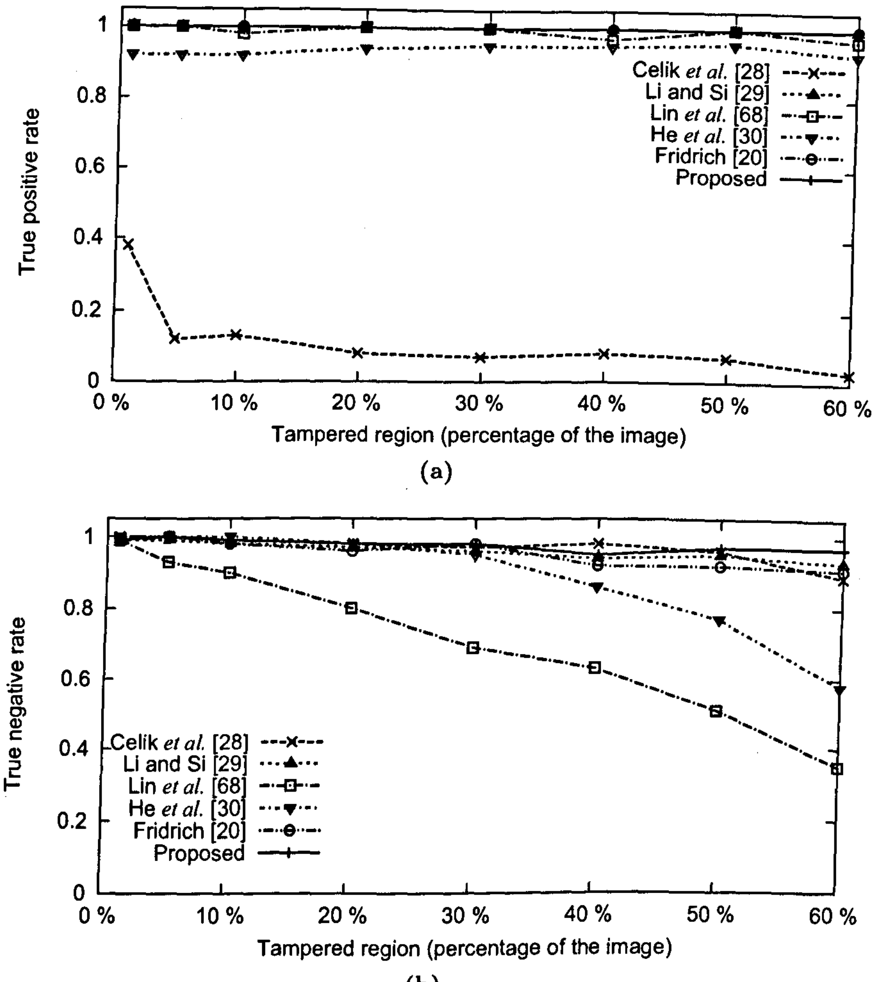
Figure 4.11: Example counterfeits. (a) inner-copy-and-paste attack where the supplanting region covers 20% of the image. (b) outer-copy-and-paste attack where the supplanting region covers 20% of the image.(c) salt-and-pepper noise (1.6%).

4.4.3 Testing localisation accuracy

Three experiments were conducted to investigate how the localisation accuracy is affected by the proportion and the dispersion of tampered pixels. The results obtained with the proposed method were compared with the schemes listed above. The following test images were used: the Waterfront and the F-16 images, in Figures 4.6(a) and 4.6(b), respectively, as well as the standard 512 × 512 images: Baboon, Peppers and Boat.

Inner-copy-and-paste attacks

Periodical duplicates of the upper-left block of pixels of every host image were used generate a tiled image. The size of the periodical duplicates was adjusted in accordance with the block-size of each method in order to replicate integer blocks. Then, a centred square of each host image was replaced with the corresponding portion of its tiled version. The square region was adjusted to contain different percentages of the total number of pixels; the tested percentages included 1%, 5% and varied from 10% to 60%, with increments of 10%. An example counterfeit generated with the Baboon image, wherein the supplanting region covers 20% of the image, is illustrated in Figure 4.11(a). The averaged simulation results are presented in Figures 4.12(a) and 4.12(b). Celik *et al.*'s detection performance is critically undermined by this attack: it failed to identify



(a)

(b)

Figure 4.12: Comparison results from the inner-copy-and-paste attacks. (a) detection performance, and (b) validation performance.

over 80% of the altered pixels when the tampered region extended beyond 5% of the image. Both Lin *et al.*'s and He *et al.*'s methods (Type 2) showed a similar behaviour: their detection performance was consistently high across the tests, while the validation performance gradually decreased in tests involving the larger tampered regions. Yet, He *et al.* managed to alleviate the problem by means of the computed erosion. Such mechanism, however, was particularly effective when the tampered region extended to up to 30% of the image. Throughout the tests, the proposed method achieved a high validation performance. The slight decrease observed when the tampered region that covered 40% of the image is because, in this case, the tampered region did not cover entire blocks of 8×8 (the block-size employed by the proposed system). Nonetheless, in the last three tests, the percentage of genuine pixels correctly validated by the proposed method increased over 2%, in comparison with Li and Si's method, and over 4%, in comparison with Fridrich's scheme.

Outer-copy-and-paste attacks

For this test, the same key was employed to watermark every single image. Then, a centred square of the watermarked version of the Boat image was replaced with the corresponding portion of each host image. The size of the supplanting region was adjusted to cover different percentages of the total number of pixels in the image; the tested percentages included 1%, 5%, as well as 10% to 60%, with increments of 10%. Figure 4.11(b) depicts an example counterfeit wherein 20% of the Boat image has been replaced with the corresponding portion of the Peppers image. The averaged results are presented in Figures 4.13(a) and 4.13(b). The localisation capabilities of methods of Type 1 were spoilt by this attack. For example, across the tests, a large percentage of the tampered pixels went unnoticed by Celik *et al.*'s and Li and Si's methods. As for methods relying on block-wise non-deterministic dependence mechanisms (Type 2), the correct validation of authentic pixels gradually decayed as the proportion of tampered pixels increased. For example, observe the behaviour exhibited by Lin *et al.*'s and He *et al.*'s methods. The erosion computed in He *et al.*'s method manages to mitigate the problem, specially when the supplanting region extended up to 30% of the image. Fridrich's and the proposed scheme achieved the better localisation. Every single tampered pixel was effectively identified, and the validation performance was consistently high across the tests. The validation performance achieved by the proposed method slightly decreased at 40% because the tampered region did not cover entire blocks of 8×8 (the block-size employed by the proposed system). Nevertheless,

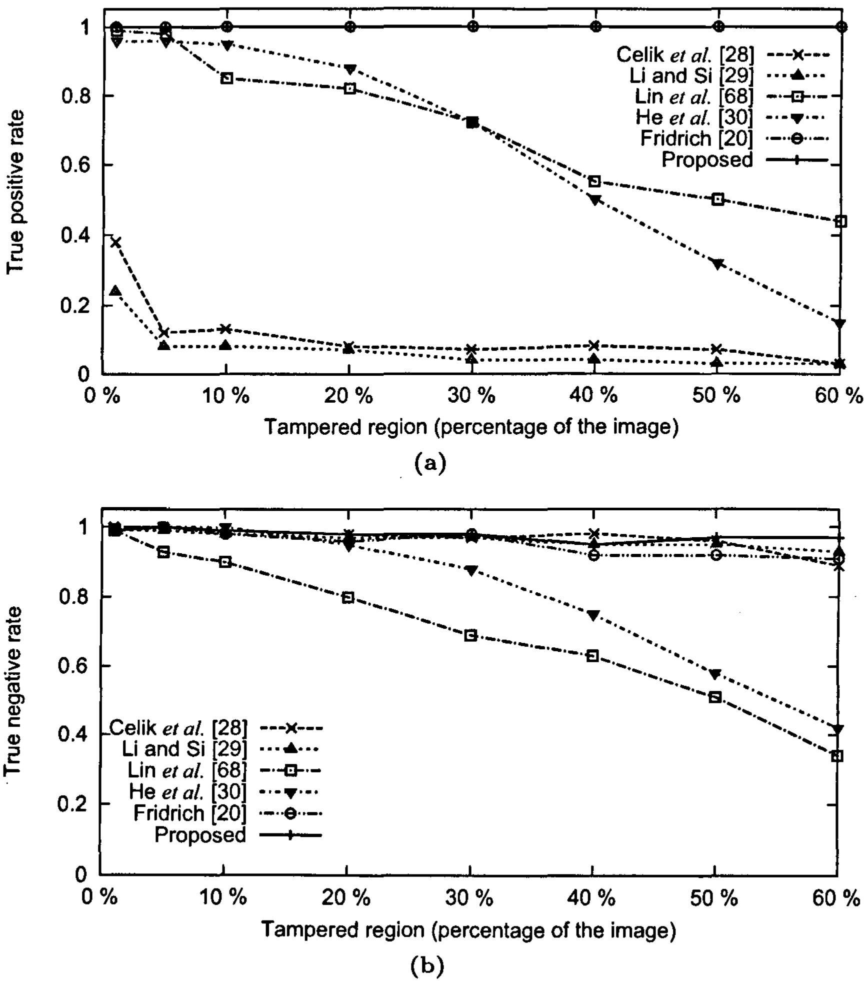
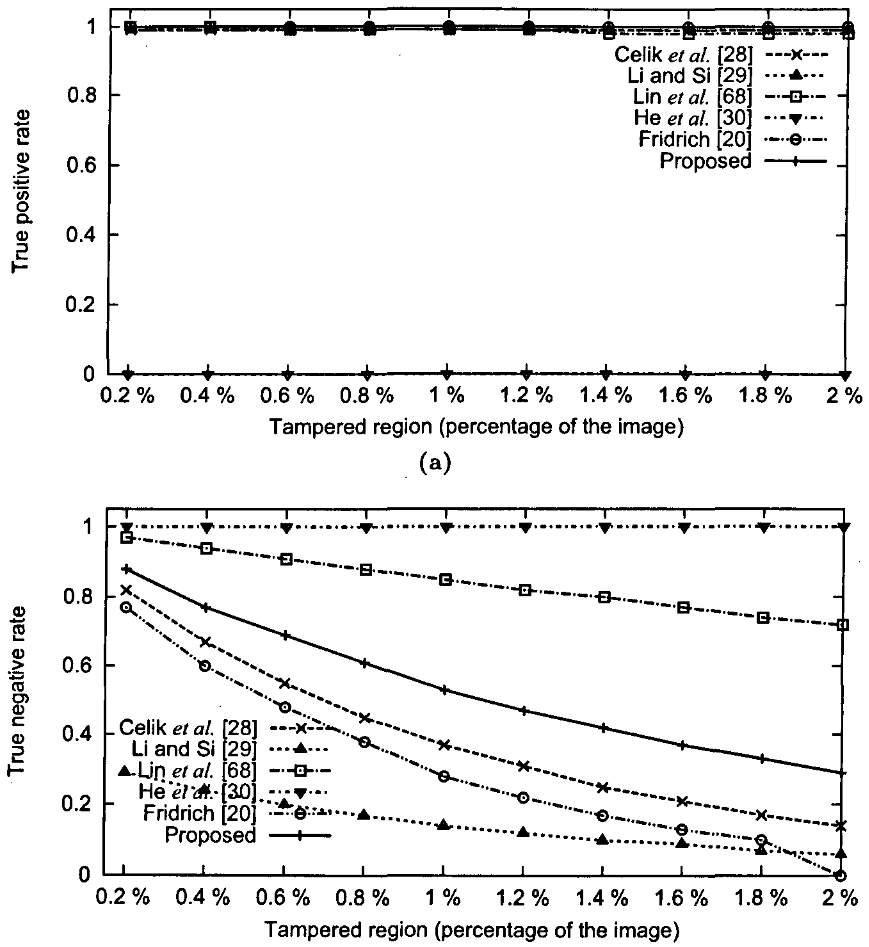


Figure 4.13: Comparison results from the outer-copy-and-paste attacks. (a) detection performance, and (b) validation performance.

in the last three tests, the proposed method managed to increase the percentage of genuine pixels correctly validated by over 5%, in comparison with Fridrich's method.

Salt-and-pepper noise attacks

The following experiment was conducted to compare the localisation performance of the methods in presence of scattered distorted pixels. Each test image was firstly watermarked and subsequently distorted to create a set of 10 forgeries; a total of 50



(b)

Figure 4.14: Comparison results from the salt-and-pepper noise attacks. (a) detection performance, and (b) validation performance.

forgeries were generated. Each forgery contained a different percentage of pixels distorted by salt-and-pepper noise, ranging from 0.2% to 2%, with increments of 0.2%. Special care was taken to ensure that the location of altered pixels was the same when testing all the methods. The averaged results, in Figures 4.14(a) and 4.14(b), show that He *et al.*'s method failed to detect the distortion in all the tests; virtually all the pixels were deemed genuine. This is a consequence of the of the erosion applied over the encoded map, which can be critical in applications where the integrity of every single bit is essential. Although Li and Si's managed to identify virtually all the tampered pixels throughout the tests, the achieved validation performance was rather poor. Fridrich's, Celik *et al.*'s and the proposed method showed a perfect detection performance across the tests. Around 80% of the authentic pixels were properly validated by the three schemes in the first experiment. However, the validation performance gradually decayed for the subsequent tests. The small block-wise used in Lin *et al.*'s scheme allowed a more accurate localisation of salt-and-pepper noise, at the expense of increasing the embedding distortion (recall Section 4.4.2). Throughout the tests, over 96% of the tampered pixels were effectively identified, and 70% of the genuine pixels were correctly validated, by Lin *et al.*'s method.

4.5 Summary and Conclusions

Most fragile watermarking methods cannot differentiate among images watermarked with the same key. Furthermore, keeping track of the key used for every single image may become unmanageable when the number of host images increase significantly – e.g. CCTV cameras. This chapter revisited Fridrich's scheme as a suitable solution for this problem, and studied its security limitations in applications where a higher localisation accuracy is needed. Additionally, a fragile watermarking as been proposed, whereby higher localisation accuracy can be achieved, while deterring brute force attacks. Unlike Fridrich's method, no duplicated data is required in the structure embedded in every block. Instead, the proposed method takes a different approach to verification, which goes through the bit strings extracted from each block to identify significant similarities, which are then used to authenticate every single block. This enables the use of longer image indices and allows automated authentication in a single stage, without compromising the security of the method.

Comparison results have been presented to evaluate the localisation performance of the proposed method and five existing methods, proposed by Celik *et al.* [28], Li and Si [29], Lin *et al.* [68], He *et al.* [30] and Fridrich [20]. The investigation has revealed some factors strongly related to the localisation capabilities of the schemes. For example, when the proportion of tampered pixels increases, the correct validation of authentic pixels decays for schemes that rely on non-deterministic block-wise dependence mechanisms (Type 2).

The localisation performance of each method may vary for different types of distortions. For example, He *et al.*'s method failed to detect and localise scattered distorted pixels, while the localisation achieved with the proposed method outperformed Celik *et al.*'s, Li and Si's, He *et al.*'s and Fridrich's schemes. Moreover, Lin *et al.*'s method managed to validate a significantly greater proportion of genuine pixels across the tests.

Schemes in the same block-wise dependence type exhibited a similar behaviour in presence of outer-copy-and-paste attacks. In fact, it was shown that this attack can jeopardise the localisation capabilities of Celik *et al.*'s and Li and Si's schemes and, in a lesser extent, Lin *et al.*'s and He *et al.*'s methods, resulting in a significant proportion of altered pixels going undetected. The consequences of deriving evidence from manipulated areas, mistakenly regarded as genuine, may be utterly critical in security-related applications, such as surveillance cameras. For these methods to overcome this problem, a different key must be used to watermark each image. As pointed out initially, however, in applications such as CCTV cameras, where millions of images can be captured daily, keeping track of the key associated with every image may be an arduous task. Fridrich's and the proposed method avoid this problem by associating a unique image index to every single image. Thus, the same key can be used to watermark numerous images without compromising the security of the verification mechanism. Nonetheless, as elucidated in Section 4.1, Fridrich's scheme becomes susceptible to brute force attacks when a small block-size, say 8 × 8, is used. Conversely, as discussed in Section 4.3.3, the proposed method renders brute force attacks computationally unfeasible when the key space, which is independent of the used block-size, is chosen to be sufficiently large.

Chapter 5

Secure Fragile Watermarking Method with Improved Self-Recovery Capabilities

5.1 Introduction

In this chapter, a fragile watermarking aimed at providing improved tampering localisation and self-recovery capabilities is presented. First, a secure block-wise mechanism, resilient to cropping, is used to localise distorted blocks of pixels. Subsequently, the tampering localisation accuracy is refined by means of an iterative pixel-wise mechanism, which estimates the original watermarked pixels in altered regions. Extensive tests, using a large data-set, are presented to show to what extent the

Section 2.5.4 reviewed watermarking methods with exact self-recovery capabilities, which are aimed at reconstructing the original watermarked content in manipulated regions. In Zhang and Wang's method [65], a pixel-wise and a block-wise mechanism are hierarchically structured to calculate the 5 MSBs of the tampered pixels. However, the block-wise method does not adopt any block-wise dependence mechanism to prevent VQ attacks. Moreover, watermarks are always embedded in the 3 LSBPs of the host images. Such lack of flexibility may render the method unsuitable for applications that require a lesser embedding distortion. Further still, not a single pixel can be restored when the image has been cropped.

Enlightened by Zhang and Wang's idea [65] of combining block-wise and pixel-wise mechanisms for tampering localisation and self-recovery, this chapter proposes a scheme with improved capabilities. Compared with their scheme, a pixel-wise method is proposed, whereby a significantly higher proportion of altered pixels can be effectively recovered, as long as the extension of tampered regions is not a very large fraction of the total pixels in the image. This is achieved by a threshold-based mechanism capable of recovering corrupted pixels, even when associated to subsets containing pixels mistakenly deemed fake by the block-wise method. The pixel-wise method can be iteratively repeated, so that a different bit-plane is watermarked in each iteration. Additionally, a variant of the block-wise method in Chapter 4 is proposed to survive cropping, while restoring the original size of a host image by incorporating an offset correction mechanism. This enables the pixel-wise method to recover pixels in the missing part. It is worth mentioning that the problem of recovering contents in regions removed by cropping has never been addressed in existing literature.

The rest of the chapter is structured as follows. The proposed scheme is detailed in Section 5.2 and its security is analysed in Section 5.3. In Section 5.4, experimental results are presented to compare the proposed method against Zhang and Wang's scheme in presence of conventional tampering and cropping. The interest in comparing with Zhang and Wang's method esteemed from the fact that it provides exact self-recovery capabilities, instead of approximate restoration capabilities as in [12, 68–72]. Furthermore, an extensive evaluation is presented to show to what extent the size of the tampered region and the proportion of tampered pixels influence the localisation and recovery performance of the two methods. Finally, the chapter is summarised and concluded in Section 5.5.

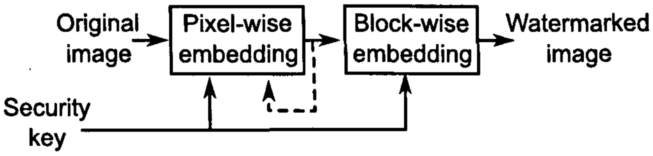


Figure 5.1: Proposed embedding phase.

5.2 Proposed Method

The proposed scheme consists of an embedding phase, which can be hard-wired in the camera, and a verification/recovery phase, which can be installed in a regular computer.

5.2.1 Embedding phase

As illustrated in Figure 5.1, this phase is comprised of the pixel-wise and the block-wise embedding methods detailed below.

Consider an $n_1 \times n_2$ grey-scale image X and denote the total number of pixels as n_X (i.e. $n_X = n_1 n_2$). Let u be the number of LSBPs that will be used to carry the computed authentication data; in practice, u is usually set to 2 or 3. The remaining u' MSBPs contain the most relevant perceptual information, which will remain intact throughout the embedding process.

Pixel-wise embedding method

In brief, the pixel-wise embedding method splits the image in subsets of pixels randomly distributed across the image. The aim is to establish a dependence mechanism among the pixels in each subset, so that any change in a pixel can be collectively detected by all the pixels associated to the same subset as the altered pixel.

As a preliminary step, the image matrix X is flattened into a $1 \times n_X$ array, denoted as $\dot{X}$. The procedure described below will be iteratively repeated $(u-1)$ times, so that a different bit-plane will be watermarked in each iteration; let $z = u$ be an integer that indicates the bit-plane to be watermarked in the first place. Every iteration involves the steps below.

1. A function, $\mathcal{S}(\cdot)$, is used to pseudo-randomly shuffle the pixels in $\dot{X}$, as $\ddot{X} = \mathcal{S}(\dot{X}, \alpha)$, where α is a pseudo-random seed. Next, $\ddot{X}$ is split into non-overlapping arrays of $1 \times m_{\text{pw}}$ pixels. To compute a different shuffle in each iteration, set $\alpha = (k + z)$, where k is a secret key. This way, every pixel is associated to $(u - 1)$ different arrays of pixels. This is beneficial for the localisation enhancement and recovery mechanisms detailed in Section 5.2.2.

2. Let $\ddot{X}_q$ be the q-th array, for $q = 1, \ldots, (n_X/m_{\text{pw}})$. For each $\ddot{X}_q$, a bit string v_q, of length m_{pw}, is computed as,

$$v_q = \mathcal{H}(k, z, q, \tilde{X}_q) \ , \tag{5.1}$$

where $\mathcal{H}(\cdot)$ is a cryptographic hash function[1], $\tilde{X}_q = 2^u \lfloor 2^{-u} \ddot{X}_q \rfloor$ and $\lfloor \cdot \rfloor$ denotes the floor function. The parameters k, z and q are included, in Equation (5.1), for security reasons. Then, form a $1 \times m_{\text{pw}}$ binary array, denoted as V_q, with the bits in v_q, and proceed to watermark the z-th bit-plane of $\ddot{X}_q$, by $\ddot{X}_q = 2^z \lfloor 2^{-z} \ddot{X}_q \rfloor + 2^{(z-1)} V_q$.

3. Once every single array has been watermarked, return all the pixels to their original location by $\dot{X} = \mathcal{S}^{-1}(\ddot{X}, \alpha)$, where $\mathcal{S}^{-1}(\cdot)$ is the inverse shuffle function, such that $\dot{X} = \mathcal{S}^{-1}(\mathcal{S}(\dot{X}, \alpha), \alpha)$. Next, if $z > 1$, reduce the value of z by one and repeat the procedure to watermark another bit-plane.

After $(u - 1)$ iterations, update X as an $n_1 \times n_2$ reshaped version of $\dot{X}$.

[1] Standard cryptographic hash functions, such as MD5 or SHA, generate strings of 128 bits or longer [132, 133]. However, since the distribution of such bit strings is virtually uniform, they can be readily trimmed off to m_{pw} bits.

Block-wise embedding method

To afford resilience to cropping and VQ attacks [3, 4], a variant of the block-wise method presented in Chapter 4 is proposed.

First, divide X into non-overlapping blocks of $m_1 \times m_2$ pixels; let X_p be the p-th block in X, for $p = 1, \ldots, (n_X/m_{\text{bw}})$, where $m_{\text{bw}} = m_1 m_2$ is the total number of pixels in each block. For each block X_p, encode an authentication bit string w_p, of length m_{bw}, as,

$$w_p = \vartheta_X \,||\, n_1 \,||\, n_2 \,||\, p \ , \tag{5.2}$$

where ϑ_X is an image index exclusively associated to X and $||$ denotes concatenation of bits. Observe that the authentication bit string, in Equation (5.2), includes the shape of the image, as well as a block index. Later on, it will be discussed how this information can provide resilience to cropping. Furthermore, note that the authentication bit strings encoded for all the blocks share a common prefix (i.e. $\vartheta_X||n_1||n_2$), which can be thought of as a unique index, whose length, in bits, will be denoted by μ_X. As in the method presented in Chapter 4, the unique index is essential for the authentication mechanism described in the forthcoming section.

Then, compute a bit string, of length m_{bw}, as,

$$d_p = \mathcal{H}(k, \hat{X}_p) \oplus w_p \ , \tag{5.3}$$

where $\hat{X}_p = 2\lfloor 2^{-1} X_p \rfloor$. Finally, with the bits in d_p, form an $m_1 \times m_2$ binary matrix D_p, which is subsequently embedded to obtain a watermarked block by, $X_p^w = \hat{X}_p + D_p$.

5.2.2 Verification and recovery phase

Figure 5.2 illustrates the three stages of this phase, which are detailed below.

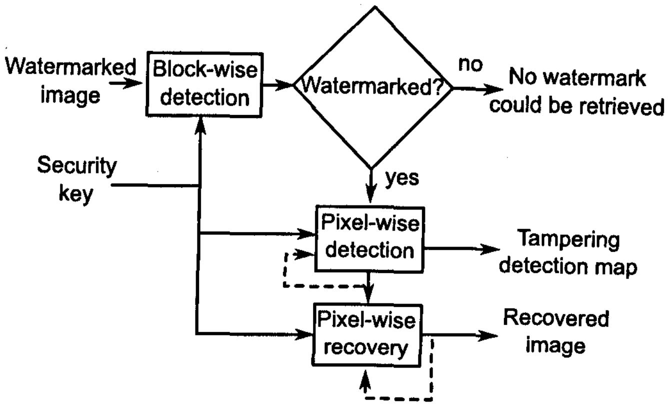


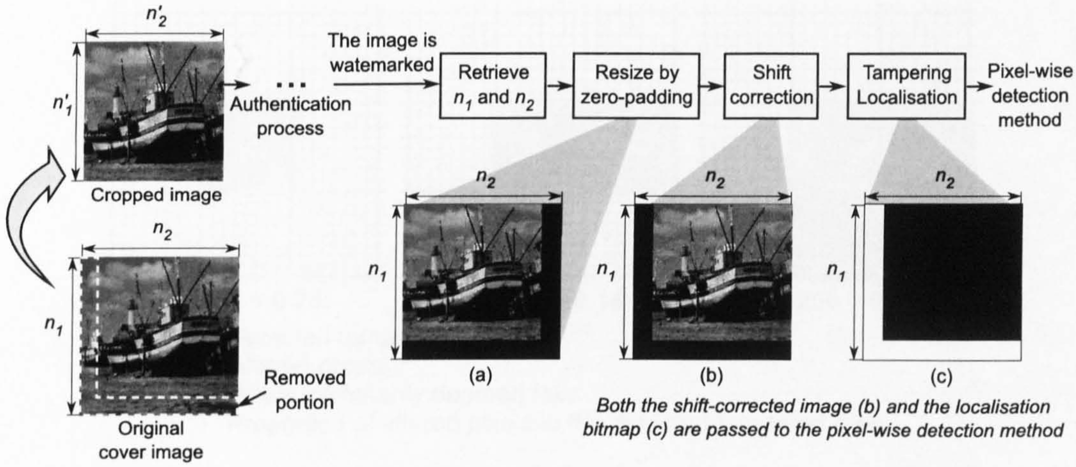
Figure 5.2: Proposed image verification and recovery phase.

Block-wise detection method

Consider an $n_1' \times n_2'$ image Y, which is divided into non-overlapping blocks of $m_1 \times m_2$ pixels. For every block Y_p, encode a bit string d_p', of length m_{bw}, with the LSB of every pixel in Y_p and compute an authentication bit string,

$$w_p' = \mathcal{H}(k', \hat{Y}_p) \oplus d_p' \ , \tag{5.4}$$

where k' is the secret key and $\hat{Y}_p = 2\lfloor 2^{-1} Y_p \rfloor$. Let $\mathcal{L}(w_p')$ be a function that retrieves the prefix formed by the μ_X left-most bits in w_p'. Additionally, let $\mathbf{A} = \{w_{a_1}', \ldots, w_{a_f}'\}$ be a set of authentication bit strings, such that $\mathcal{L}(w_{a_1}') = \ldots = \mathcal{L}(w_{a_f}')$. That is, all the bit strings in $\mathbf{A}$ contain the same unique index. If Y is a watermarked, probably tampered, image and $k' = k$, it is expected that most of the extracted authentication bit strings will contain the same unique index. Hence, if $|\mathbf{A}|$ – i.e. the cardinality of the set $\mathbf{A}$ – is greater than a predefined threshold τ_{bw}, the image is regarded as watermarked. Otherwise, if $|\mathbf{A}| \leq \tau_{\text{bw}}$, generate a set of $m_1 m_2$ different shifted versions of Y. In a shifted version, all the pixels in Y are displaced λ_1 rows and λ_2 columns, where $-m_1 < \lambda_1 \leq 0$ and $-m_2 < \lambda_2 \leq 0$. Every shifted version is analysed as described above. This way, the system can identify host images whose left and/or upper-most



Both the shift-corrected image (b) and the localisation bitmap (c) are passed to the pixel-wise detection method

Figure 5.3: Essential steps to cope with cropping.

edges have been removed by cropping. If none of the shifted versions was reckoned to be watermarked, the detection process is terminated altogether.

If Y is deemed watermarked, proceed to determine whether it has been cropped, as illustrated in Figure 5.3. First, retrieve n_1 and n_2 from the prefix with higher occurrence. If it has been cropped (i.e. if $n'_1 < n_1$ and/or $n'_2 < n_2$), reshape Y into $n_1 \times n_2$ by adding rows/columns of zeros as exemplified in Figure 5.3(a). At this point, the size of Y has been restored, but its content may appear shifted from its original location. This situation would hinder the pixel-wise method, as it depends on the scanning order. To overcome this problem, the block index retrieved from every authentication bit string, in $\mathbf{A}$, is used to estimate a common displacement. Let $\mathcal{I}(w'_p)$ be a function that retrieves the block index from w'_p and $\mathbf{B} = \{w'_{b_1}, \ldots, w'_{b_g}\}$, $\mathbf{B} \subseteq \mathbf{A}$, be a set, such that $\delta = \mathcal{I}(w'_{b_1}) - b_1 = \ldots = \mathcal{I}(w'_{b_1}) - b_g$. The value of δ indicates the number of block slots the content has to be shifted. Therefore, the image content has to be displaced $m_1\delta_1$ rows and $m_2\delta_2$ columns, where $\delta_1 = \lfloor m_2\delta/n_2 \rfloor$, and $\delta_2 = (\delta - n_2\delta_1/m_2)$. An example output of the shift correction is illustrated in Figure 5.3(b). Finally, as illustrated in Figure 5.3(c), an $n_1 \times n_2$ bitmap M, split into blocks of $m_1 \times m_2$ pixels, is encoded for tampering localisation: the block index retrieved from every bit string

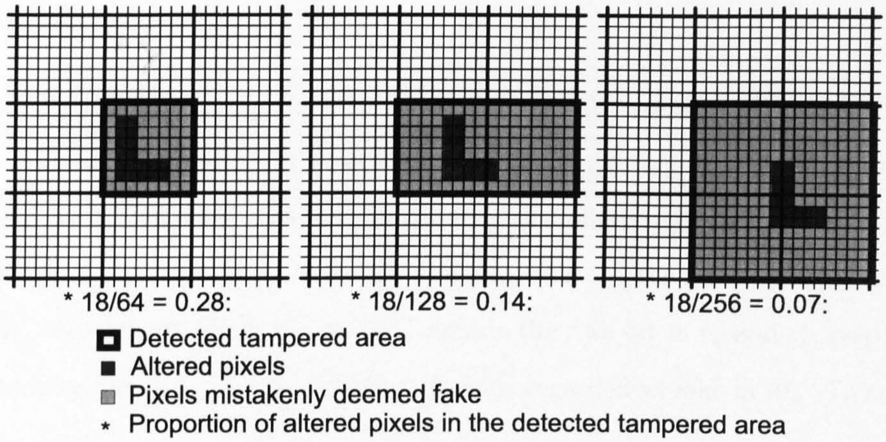

* 18/64 = 0.28:
* 18/128 = 0.14:
* 18/256 = 0.07:
▣ Detected tampered area
■ Altered pixels
▨ Pixels mistakenly deemed fake
* Proportion of altered pixels in the detected tampered area

Figure 5.4: Tampering localisation of the block-wise method.

in **B** is used to localise an authentic block, which is set to zero; the rest of the blocks – i.e. altered blocks – are set to one.

Pixel-wise detection method

The block-wise detector manages to localise blocks containing at least one corrupted pixel. However, the area regarded as tampered may contain genuine pixels, as well as pixels whose u' MSBs remain intact. Those pixels will be considered *mistakenly deemed fake*. This is illustrated in Figure 5.4 using three different examples. Observe that, even though the number of altered pixels is the same in the three cases, the detected tampered area varies significantly. The pixel-wise detection method described below is aimed at validating some of those pixels to enhance the localisation.

Let $\dot{Y}$ and $\dot{M}$ be flattened versions of the matrices Y and M, respectively. As in the embedding method, the procedure will be iteratively repeated $(u-1)$ times; let $z = u$ be the bit-plane to be analysed in the first iteration. Each iteration is comprised of the following steps.

1. Compute the shuffled versions of $\dot{Y}$ and $\dot{M}$, by $\ddot{Y} = \mathcal{S}(\dot{Y}, \alpha')$, and $\ddot{M} = \mathcal{S}(\dot{M}, \alpha')$, where $\alpha' = (k' + z)$. Then, divide $\ddot{Y}$ and $\ddot{M}$ into non-overlapping arrays of $1 \times m_{\text{pw}}$

pixels.

2. Let $\ddot{Y}_q$ and $\ddot{M}_q$ denote the q-th array in $\ddot{Y}$ and $\ddot{M}$, respectively. For each array, encode a bit string $\breve{v}_q$, of length m_{pw}, by concatenating the bits in the z-th bit-plane of $\ddot{Y}_q$. Besides, compute the bit string of length m_{pw}, as $v'_q = \mathcal{H}(k', z, q, \tilde{Y}_q)$, where $\tilde{Y}_q = 2^u \lfloor 2^{-u} \ddot{Y}_q \rfloor$. Let $\ddot{Y}_{q,r}$ and $\ddot{M}_{q,r}$ denote the r-th element in $\ddot{Y}_q$ and $\ddot{M}_q$, respectively, while $\breve{v}_{q,r}$ and $v'_{q,r}$ denote the r-th bit in $\breve{v}_q$ and v'_q, respectively. Additionally, let c_q be the number of pixels regarded as fake in $\ddot{M}_q$. To constrain the possible increase of false negatives, the analysis of the current array is skipped if $c_q > \tau_{\text{pw}}$, where τ_{pw} is a predefined threshold (see Section 5.3).

   If the c_q pixels were mistakenly deemed fake, one of the following two scenarios must apply. 1) $v'_q = \breve{v}_q$. This scenario obviously indicates that every single pixel in $\ddot{Y}_q$ is genuine. 2) $v'_q \neq \breve{v}_q$, but, for every $\ddot{M}_{q,r} = 0$ it follows that $v'_{q,r} = \breve{v}_{q,r}$. This scenario reveals that the u LSBs of at least one of the c_q pixels have changed, but their u' MSBs remain intact. Therefore, if any of these two scenarios takes place, the c_q pixels are validated by setting $\ddot{M}_q$ to zero.

   Because of the properties of cryptographic functions [133], any change on the u' MSBPs of $\ddot{Y}_q$ will result in a bit string v'_q that obeys uniform distribution. Hence, the probability that one of the two scenarios above will apply is very low (see Section 5.3). Thus, if none of those scenarios applied, the c_q pixels are deemed fake and $\ddot{M}_q$ is kept intact.

3. Once every array has been analysed, return all the pixels in $\ddot{M}$ to their original location by $\dot{M} = \mathcal{S}^{-1}(\ddot{M}, \alpha')$. Then, reduce the value of z by one and repeat the procedure if $z > 1$.

Observe that, in a single iteration, a pixel mistakenly deemed fake can be successfully validated, unless it belongs to an array containing, at least, one corrupted pixel. Nonetheless, because of the iterative procedure, every pixel will be associated to $(u-1)$

different arrays. Thus, a pixel mistakenly deemed fake has $(u-1)$ chances of being validated.

Pixel-wise recovery method

This process is also iteratively repeated $(u-1)$ times by following the steps below; let $z = u$ be the bit-plane to be analysed in the first iteration. Assuming that $\dot{Y}$ and $\dot{M}$ are still the flattened versions of Y and M, respectively, the steps below are performed in every iteration.

1. Compute $\ddot{Y} = \mathcal{S}(\dot{Y}, \alpha')$ and $\ddot{M} = \mathcal{S}(\dot{M}, \alpha')$, and then divide $\ddot{Y}$ and $\ddot{M}$ into non-overlapping arrays of $1 \times m_{\text{pw}}$ pixels.

2. For every array, let $\ddot{M}_{q,c_1}, \ldots, \ddot{M}_{q,c_q}$ denote pixels set to one[2] in $\ddot{M}_q$. To limit the possibility of estimating an incorrect pixel value, the recovery mechanism skips the current array if $c_q > \tau_{\text{pw}}$ (see Section 5.3). Encode a bit string $\breve{v}_q$, by concatenating the bits in the z-th bit-plane of $\ddot{Y}_q$.

   First, assume that only $\ddot{Y}_{q,c_1}$ has undergone changes in its u' MSBs. Then, conduct exhaustive trials $\ddot{Y}_{q,c_1} = 0(2^u), 1(2^u), \ldots, (2^{u'} - 1)(2^u)$, computing $v'_q = \mathcal{H}(k', z, q, \tilde{Y}_q)$ every time, to find a possible *valid pixel value*, such that for every $\ddot{M}_{q,r} = 0$ it follows that $v'_{q,r} = \breve{v}_{q,r}$. An identical procedure is carried out for the rest of the possibly tampered pixels in the array, namely $\ddot{Y}_{q,c_2} \ldots \ddot{Y}_{q,c_q}$. If only one, out of the total $2^{u'} c_q$ attempts, resulted in a valid pixel value, the array is regarded as successfully restored. Thus, the altered pixel is replaced with the valid pixel value and $\ddot{M}_q$ is set to zero to enhance the localisation bitmap further. If none or more than one valid pixel value were found, the array cannot be restored and hence both $\ddot{Y}_q$ and $\ddot{M}_q$ are kept intact.

3. Once every array has been analysed, return the pixels in $\ddot{Y}$ and $\ddot{M}$ to their original

[2] In [65], the case where subsets contain more than one pixel regarded as altered by the block-wise method was not considered.

location, i.e. $\dot{Y} = \mathcal{S}^{-1}(\ddot{Y}, \alpha')$ and $\dot{M} = \mathcal{S}^{-1}(\ddot{M}, \alpha')$. Next, reduce the value of z by one and repeat the procedure if $z > 1$.

After $(u-1)$ iterations, both $\dot{M}$ and $\dot{Y}$ are reshaped, as $n_1 \times n_2$ arrays, to update M as the final localisation bitmap and obtain the final recovered image Y^r.

Note that, in a single iteration, an altered pixel can be effectively recovered, unless it belongs to an array containing another corrupted pixel. Nevertheless, in accordance with the algorithm above, every pixel will be associated to $(u-1)$ different arrays. Thus, an altered pixel has $(u-1)$ chances of being recovered.

5.3 Analysis

In the following analysis, it will be assumed that the output of the hash $\mathcal{H}(\cdot)$ is drawn from a uniform distribution, as in the case of cryptographic hash functions – e.g. the standard MD5 or SHA [132, 133].

5.3.1 Distortion

To estimate an approximate of the distortion induced to host images, as part of the embedding phase, it will be assumed that all the bits in the original u-th bit-planes will be changed with the same probability. Thus, the MSE can be calculated as,

$$\text{MSE} = \frac{1}{2^{2u}} \sum_{i=0}^{2^u-1} \sum_{j=0}^{2^u-1} (i-j)^2 \ . \tag{5.5}$$

Therefore, it follows that the approximate PSNR is,

$$\text{PSNR} \approx 10 \log_{10} \left(\frac{2^{2u} \max_X^2}{\sum_{i=0}^{2^u-1} \sum_{j=0}^{2^u-1} (i-j)^2} \right) \ . \tag{5.6}$$

5.3.2 Security

The block-wise method is accountable for the overall security of the proposed scheme. The reader is, therefore, referred to the security analysis presented in Section 4.3.2, recalling that the unique index can be retrieved from the μ_X left-most bits of the authentication bit string extracted from every genuine block.

5.3.3 False validation

Recall that the pixel-wise detection method can be thought of as a localisation enhancement process, which is aimed at validating possible genuine pixels that were mistakenly deemed fake by the block-wise method. To determine the likelihood that an altered pixel will be regarded as genuine (i.e. validated), recall the notation in 5.2.2. For the sake of simplicity, consider the typical cases where $u = 2$ and $u = 3$ (indeed, the embedding distortion induced when using $u > 3$ may be inadequate for most applications). Consider an arbitrary tampered pixel that belongs to an array, say $\ddot{Y}_{a_1}$, in the first iteration, and to another array, say $\ddot{Y}_{a_2}$, in a possible second iteration (when $u = 3$). Let c_{a_i} denote the total number of pixels marked as altered in $\ddot{M}_{a_i}$ (for $i = 1, 2$), provided that $1 \leq c_{a_i} \leq \tau_{\text{pw}}$. Additionally, let E_{a_i} be the event that the authentication bits computed from the $(m_{\text{pw}} - c_{a_i})$ pixels marked as genuine, in $\ddot{M}_{a_i}$, match the retrieved bits. The probability that E_{a_i} will occur is $\mathcal{P}_{E_{a_i}} = 2^{-(m_{\text{pw}} - c_{a_i})}$. Since E_{a_1} and E_{a_2} are not mutually exclusive, the probability that the altered pixel will be mistakenly regarded as genuine is,

$$\mathcal{P}_{\text{fn}}(E_{a_1} \cup E_{a_2}) = \mathcal{P}_{E_{a_1}} + \mathcal{P}_{E_{a_2}} - \mathcal{P}_{E_{a_1} E_{a_2}} \ , \tag{5.7}$$

where $\mathcal{P}_{E_{a_1} E_{a_2}} = 2^{-(2m_{\text{pw}} - c_{a_1} - c_{a_2})}$ is the probability that E_{a_1} and E_{a_2} will occur in both iterations.

5.3.4 False recovery

A similar reasoning as above will be followed to determine the likelihood that an erroneous pixel value will be estimated by the recovery method. Consider an arbitrary tampered pixel that belongs to an array, say $\ddot{Y}_{b_1}$, in the first iteration, and a different array, say $\ddot{Y}_{b_2}$, in a possible second iteration. Let c_{b_i} denote the number of altered pixels in $\ddot{Y}_{b_i}$ (for $i = 1, 2$), provided that $2 \leq c_{b_i} \leq \tau_{\mathrm{pw}}$. The recovery method will exhaustively try $2^{u'}$ different values for each one of the c_{b_i} pixel locations marked as tampered in $\ddot{M}_{b_i}$ (recall Section 5.2.2). Let E_{b_i} be the event that, for a single trial, the authentication bits computed from the $(m_{\mathrm{pw}} - c_{b_i})$ genuine pixels match the retrieved bits. The probability of success of a single trial is $\mathcal{P}_{E_{b_i}} = 2^{-(m_{\mathrm{pw}} - c_{b_i})}$. Denoting, as $\mathcal{X}_{b_i}$, the total number of occurrences of E_{b_i}, the probability that only one trial, out of the total $2^{u'} c_{b_i}$, will succeed can be expressed as,

$$\mathcal{P}_{b_i}(\mathcal{X}_{b_i} = 1) = \binom{2^{u'} c_{b_i}}{1} \mathcal{P}_{E_{b_i}} (1 - \mathcal{P}_{E_{b_i}})^{2^{u'} c_{b_i} - 1} \ . \tag{5.8}$$

Thus, the probability of recovering an erroneous pixel value is,

$$\mathcal{P}_{\mathrm{fr}}(E_{b_1} \cup E_{b_2}) = \mathcal{P}_{b_1} + \mathcal{P}_{b_2} - \mathcal{P}_{b_1 b_2} \ , \tag{5.9}$$

where,

$$\mathcal{P}_{b_1 b_2}(\mathcal{X}_{b_1}{=}1, \mathcal{X}_{b_2}{=}1) = \binom{2^{u'} c_{b_1} c_{b_2}}{1} \mathcal{P}_{E_{b_1} E_{b_2}} (1 - \mathcal{P}_{E_{b_1} E_{b_2}})^{2^{u'} c_{b_1} c_{b_2} - 1} \ , \tag{5.10}$$

and $\mathcal{P}_{E_{b_1} E_{b_2}} = 2^{-(2m_{\mathrm{pw}} - c_{b_1} - c_{b_2})}$ is the probability that E_{b_1} and E_{b_2} will occur in both iterations.

5.4 Experimental Results

The proposed system was tested for watermarks embedded in 2 and 3 LSBPs in the images (i.e. using $u = 2$ and $u = 3$, respectively). In both cases, the block-wise method employed blocks of $8(m_1) \times 8(m_2)$ pixels and the pixel-wise method employed arrays of $1 \times 16(m_{\text{pw}})$ pixels. The same experiments were carried out with Zhang and Wang's method [65], using the set-up suggested by the authors. It is important to stress that Zhang and Wang's method always embeds watermarks in the 3 LSBPs. The interest in comparing the proposed method with such scheme arises from the fact that it provides exact self-recovery capabilities, instead of approximate restoration capabilities as in [12, 68–72].

To evaluate the localisation performance, both the accuracy (ACC) and the false positive rate (FPR) were calculated as,

$$\text{ACC} = \frac{M \cap G}{G}, \text{ and, } \text{FPR} = \frac{M \cup G}{G} - 1 \ ,$$

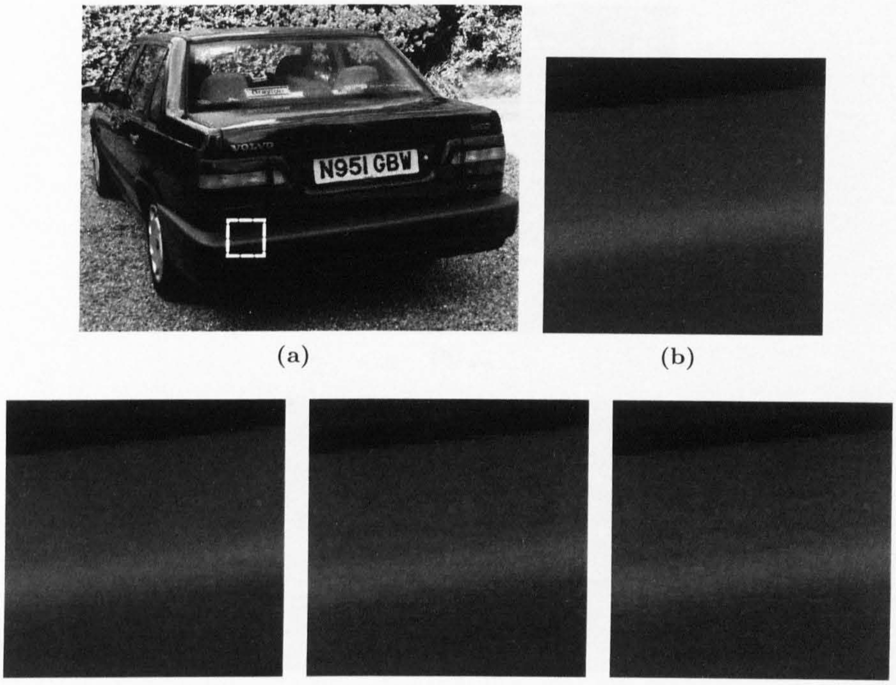
where M is the tampering localisation bitmap, encoded by the pixel-wise detection method and G is the ground truth. Observe that an ideal detection would simultaneously render ACC=1 and FPR=0.

To evaluate the self-recovery performance, let x denote the percentage of pixels that could not be recovered, which is calculated out of the total number pixels initially regarded as altered by the block-wise method[3]. In these results, it will be reported (100-x)%, which includes both the successfully validated and the successfully recovered pixels.

5.4.1 Examining the quality of the host images

The 600×800 test image used in these experiments is shown in Figure 5.5(a), and a close-up of the region bounded by the dashed square is presented in Figure 5.5(b).

[3] This is a fair comparison, as the tested methods were set up to use the same block-size (8×8).



(a)

(b)

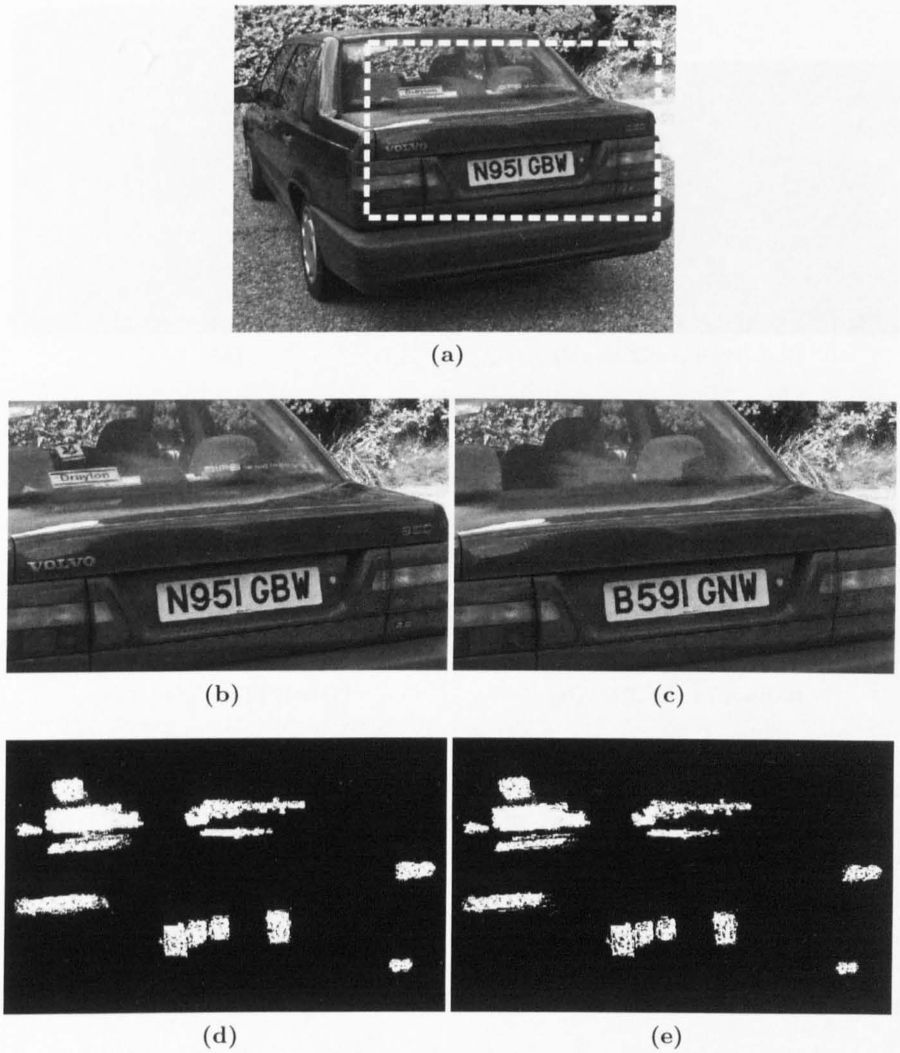
(c) PSNR=44.2 dB

(d) PSNR=37.8 dB

(e) PSNR=37.8 dB

Figure 5.5: Examining the quality of the host images. (a) Original 600 × 800 image. (b) Close-up of the region bounded by the dashed square in (a). Close-ups of the images watermarked with the proposed method: (d) using 2 LSBPs; (e) using 3 LSBPs. Close-up of the image watermarked with Zhang and Wang's scheme (3 LSBPs).

The zoomed area was specifically chosen to illustrate the impact of the watermarks on the quality of the host images, which can be perceived more clearly in regions of uniform pixel values. Figure 5.5(c) shows the close-up of the image watermarked in its 2 LSBPs with the proposed method. The watermark introduced some amount of distortion (PSNR=4.1 dB), which can be barely noticed when compared with Figure 5.5(b). On the other hand, the close-ups of the images watermarked in their 3 LSBPs, using the proposed or Zhang and Wang's scheme, are shown in Figures 5.5(d) and 5.5(e), respectively. In both cases, the PSNR values were assessed to be 37.8 dB. The distortion introduced by the watermarks produces coarser changes in the intensity, which can be noticed in the way from the black regions to the brighter regions of the images.



(a)

(b)

(c)

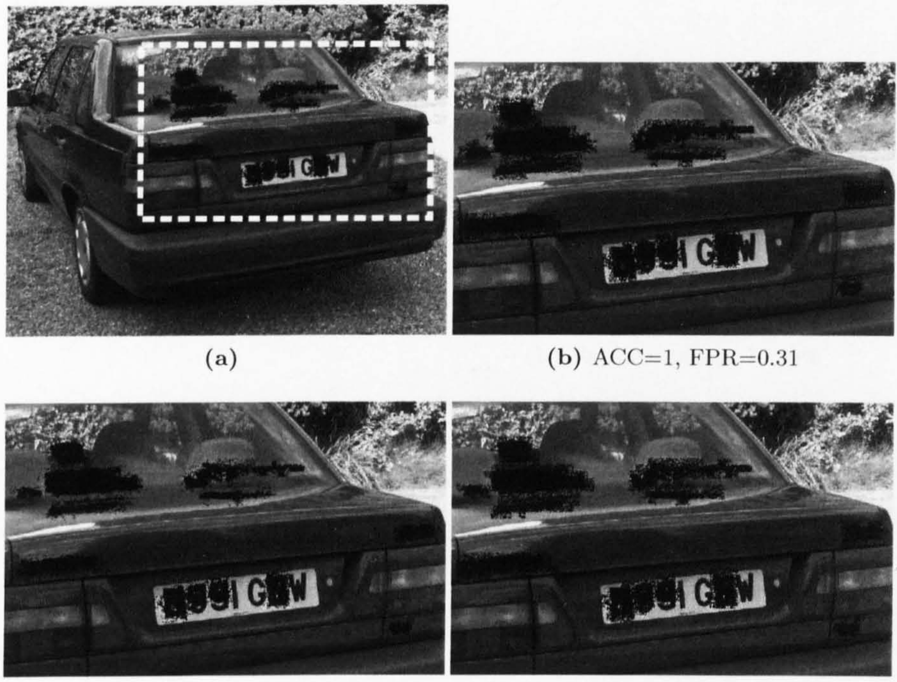
(d)

(e)

Figure 5.6: Conventional tampering. (a) Original image. (b) Close-up of the region bounded by the dashed rectangle in (a). (c) Tampered host image (close-up). (d) Pixels that have undergone changes in their 6 MSBs (close-up). (e) Pixels that have undergone changes in their 5 MSBs (close-up).

5.4.2 Conventional tampering

The three host images were manipulated to form identical forgeries. The vehicle registration plate was altered from "N95I GBW" to "B59I GNW". Additionally, the labels, "VOLVO", "850" and "2.5", on the boot, as well as the stickers on the rear window, were concealed, while the rest of the pixels remained intact. To ease visual inspection,



(a)

(b) ACC=1, FPR=0.31

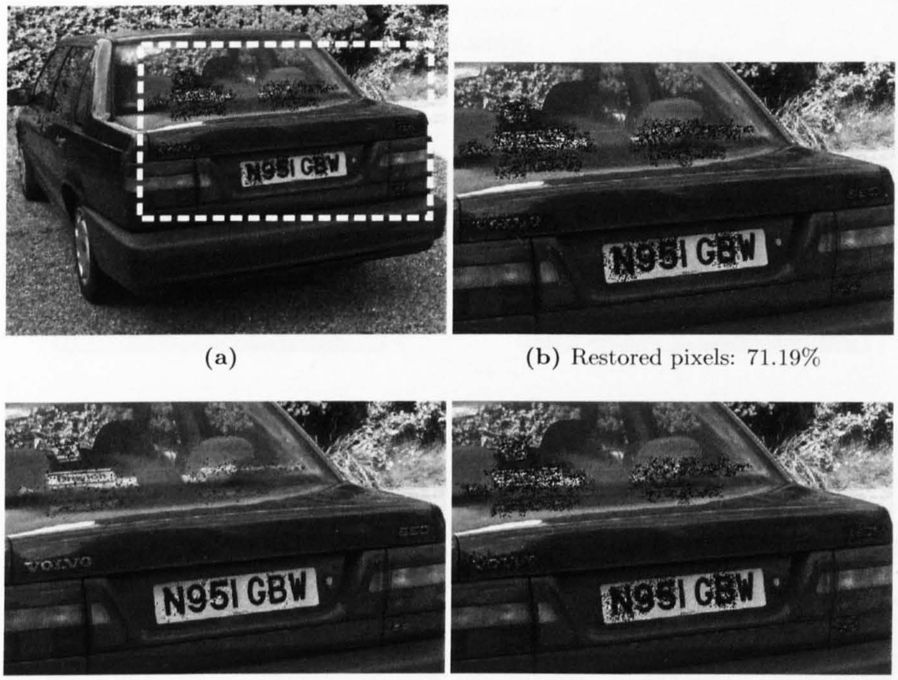
(c) ACC=1, FPR=0.11

(d) ACC=1, FPR=0.40

Figure 5.7: Localisation of conventional tampering. (a) Proposed method using 2 LSBPs. (b) Close-up of the region enclosed in (a). (c) Proposed method using 3 LSBPs (close-up). (d) Zhang and Wang's method, which uses 3 LSBPs (close-up).

close-ups of the original image and the forgery are shown in Figs. 5.6(b) and 5.6(c), respectively. The pixels that experienced changes in their 6 MSBs (13,381 in total) are depicted as white spots in Figure 5.6(d). This is, in fact, a close-up of the ground truth used to compute the results, when the watermarks were embedded in the 2 LSBPs of the image. Figure 5.6(e) depicts the pixels that have undergone changes in their 5 MSBs (11,709 in total). This is a close-up of the ground truth employed to evaluate the methods that embedded watermarks in the 3 LSBPs of the image.

In the localisation results, summarised in Figure 5.7, pixels regarded as fake are depicted as black spots. Figures 5.7(b) and 5.7(c) show the results obtained with the proposed scheme after embedding the watermark in the 2 LSBPs and 3 LSBPs, respectively, while Figure 5.7(d) shows the results obtained with Zhang and Whang's method. Observe that Zhang and Wang's method failed to validate a higher proportion



(a)

(b) Restored pixels: 71.19%

(c) Restored pixels: 96.12%

(d) Restored pixels: 72.58%

Figure 5.8: Results of self-recovery of conventional tampering. (a) Proposed method using 2 LSBPs. (b) Close-up of the region enclosed in (a). (c) Proposed method using 3 LSBPs (close-up). (d) Zhang and Wang's method, which uses 3 LSBPs (close-up).

of pixels (FPR=0.40) than the proposed scheme (FPR=0.31 and FPR=0.11). Yet, in the three cases, every tampered pixel was correctly identified (ACC=1).

Figure 5.8 summarises the self-recovery results, where pixels that could not be recovered are depicted as black spots. The percentage of pixels restored/validated by the proposed method, when using 2 LSBPs (Figure 5.8(b)), is comparable to the percentage of pixels restored/validated by Zhang and Wang's method (Figure 5.8(d)). Even better is the fact that, as observed in Figure 5.8(c), the proposed system managed to recover/validate over 96% of the pixels, when using the same number of LSBPs for watermarking as in Zhang and Wang's method (3 LSBPs).

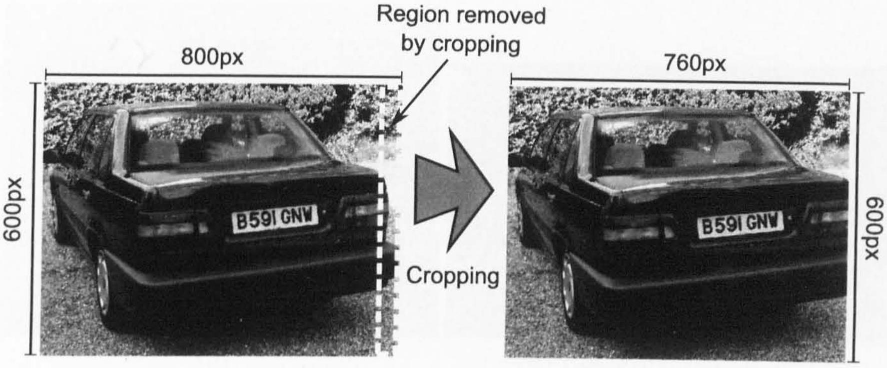
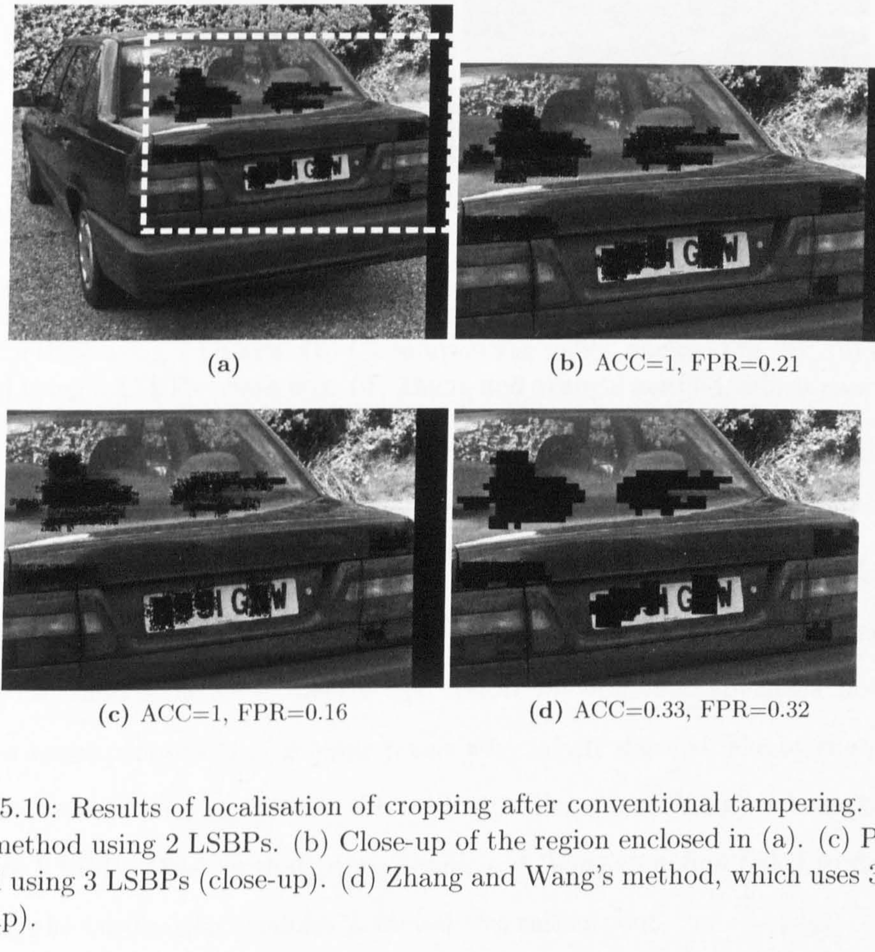


Figure 5.9: Test for conventional tampering plus cropping.



(a)

(b) ACC=1, FPR=0.21

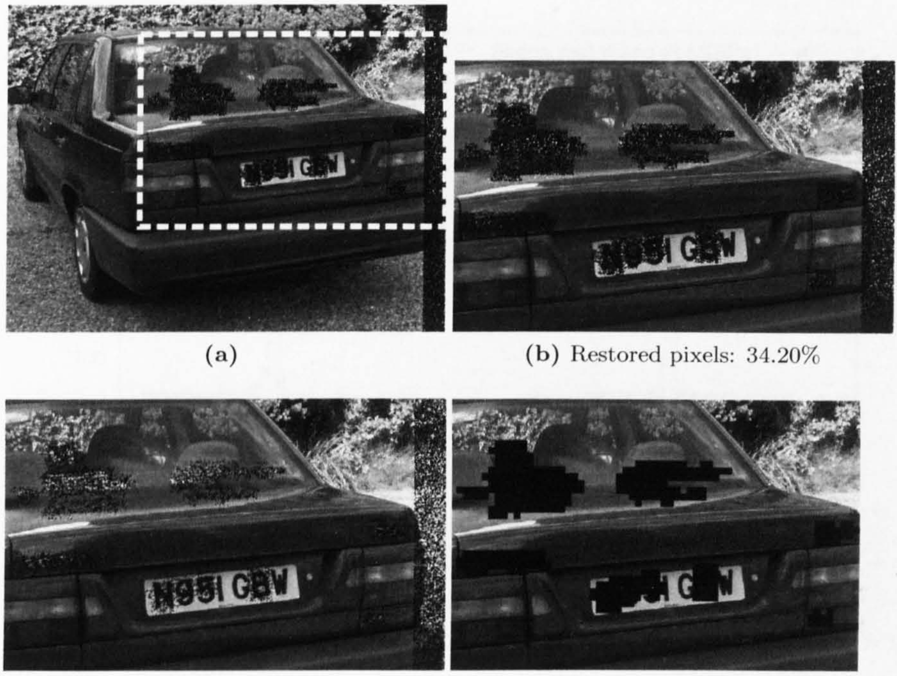
(c) ACC=1, FPR=0.16

(d) ACC=0.33, FPR=0.32

Figure 5.10: Results of localisation of cropping after conventional tampering. (a) Proposed method using 2 LSBPs. (b) Close-up of the region enclosed in (a). (c) Proposed method using 3 LSBPs (close-up). (d) Zhang and Wang's method, which uses 3 LSBPs (close-up).

5.4.3 Tampering detection in presence of cropping

As illustrated in Figure 5.9, the 40 right-most columns of the forgeries generated in the previous experiment were removed by cropping, thereby changing the dimensions



(a)

(b) Restored pixels: 34.20%

(c) Restored pixels: 63.45%

(d) Restored pixels: 0%

Figure 5.11: Results of localisation of cropping after conventional tampering. (a) Proposed method using 2 LSBPs. (b) Close-up of the region enclosed in (a). (c) Proposed method using 3 LSBPs (close-up). (d) Zhang and Wang's method, which uses 3 LSBPs (close-up).

of the image forgeries from 600 × 800 to 600 × 760.

The localisation results acquired with the proposed method, using 2 LSBPs and 3 LSBPs, are summarised in Figures 5.10(b) and 5.10(c), respectively. Note that the proposed method managed to identify and restore the original shape of the host image. Hence, a lesser proportion of genuine pixels were falsely deemed fake by the proposed system, compared with the results obtained with Zhang and Wang's scheme presented in Figure 5.10(d). Observe that, since Zhang and Wang's method failed to detect the cropping, the localisation accuracy achieved was rather poor.

Results show that the proposed method managed to restore almost 35% of the pixels, when using 2 LSBPs (Figure 5.11(b)) and over 60% when using 3 LSBPs (Figure 5.11(c)). In fact, observe that the original plate number is now discernible in the

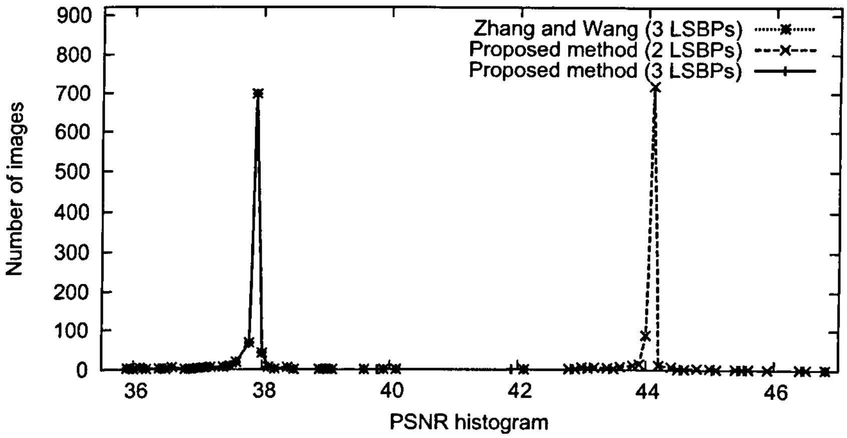


Figure 5.12: Comparison of embedding distortion.

recovered image displayed in Figure 5.11(c). Conversely, Zhang and Wang's method lost synchronisation with the watermark. Therefore, it failed to recover/validate a single pixel, as shown in Figure 5.11(d).

5.4.4 Extensive performance comparison

All the experiments presented in this section were conducted on 900 images, sized 480 × 640, in the Caltech-256 data-set [137].

Embedding distortion

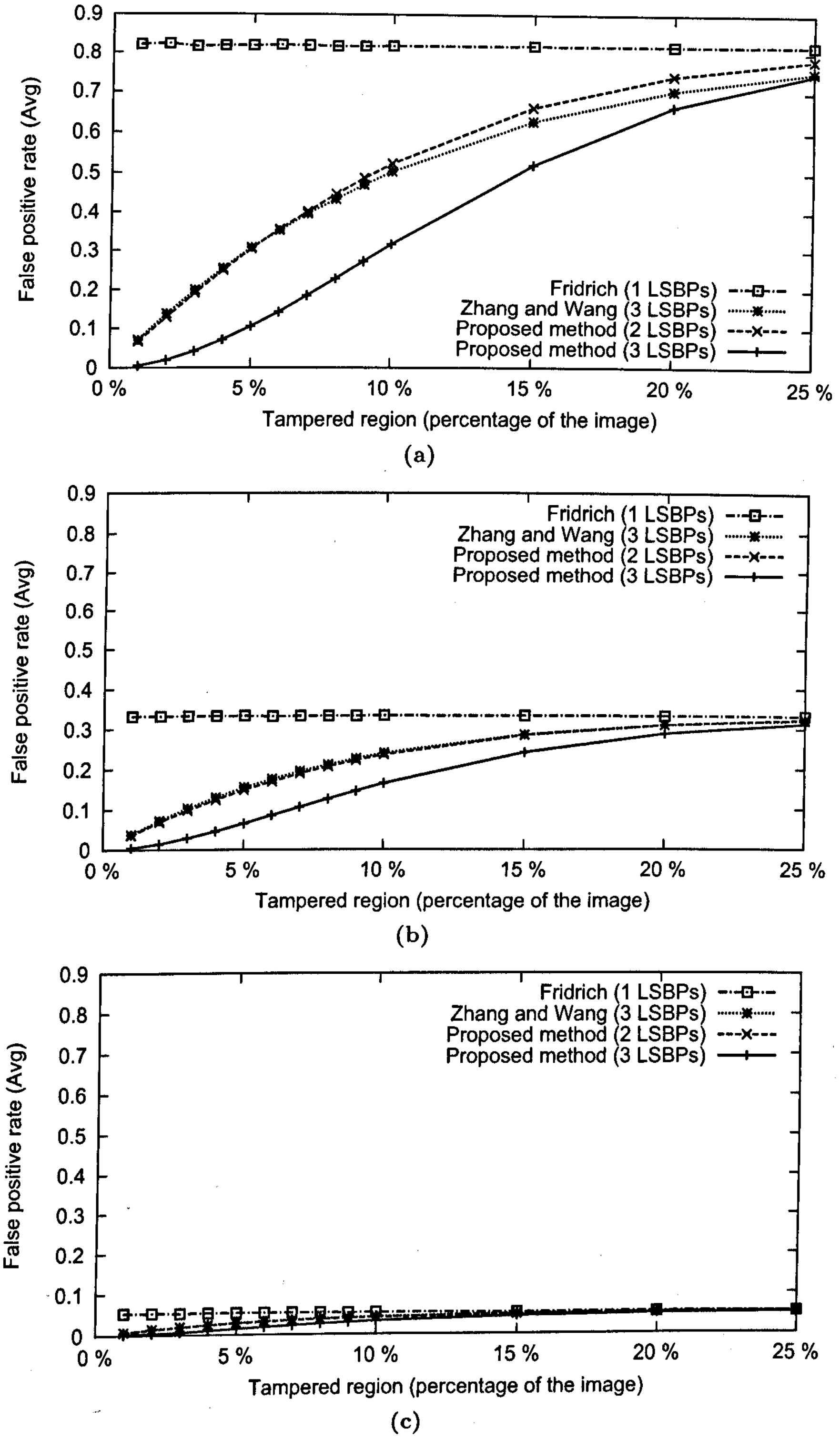
To compare the two methods in terms of embedding distortion, the PSNR between every test image and its corresponding watermarked version was computed. Figure 5.12 shows a histogram generated with the resulting PSNR values. On average, the PSNR was assessed to be 44 dB when the proposed system allocated the watermarks in the 2 LSBPs of the host images. The embedding distortion increased when the watermarks were embedded in the 3 LSBPs, either with the proposed method or with Zhang and Wang's scheme. In these cases, the PSNR values computed from 92% of the host images laid below 38 dB (the average PSNR was assessed to be 37.8 dB).

Localisation and self-recovery

To investigate to what extent the size of the tampered area localised by the block-wise method and the proportion of altered pixels within that area influence the performance of the schemes, the following experiment was formulated. In each watermarked image, a rectangular region was adjusted to cover integer blocks of 8×8 pixels, so that its area covered a percentage of the total number of pixels in the image. The tested percentages varied from 1% to 10%, with unitary increments, as well as 15%, 20% and 25%. Within the selected area, only a proportion of the pixels was tampered, by flipping their MSB. The tested proportions were: 0.55, 0.75 and 0.95.

The localisation results corresponding to the three tested proportions of altered pixels – namely, 0.55, 0.75 and 0.95 – are summarised in Figures 5.13(a), 5.13(b) and 5.13(c), respectively. The results obtained with Fridrich's method [20] were merely included as a reference; in this method, watermarks are embedded in the LSBP of the image. In most of the tests, the difference between the localisation performance achieved with the proposed method, using 2 LSBPs, and Zhang and Wang's scheme is rather marginal, even though the later uses 3 LSBPs for watermarking. Nonetheless, a higher localisation accuracy was attained when the proposed system allocated the watermark in the 3 LSBPs, particularly in forgeries containing greater proportions of altered pixels. When the tampered areas extended beyond 20% of the image, the performance exhibited by the proposed and Zhang and Wang's method is comparable and shows a clear trend towards the performance achieved with Fridrich's scheme. The same trend can be observed for the three tested proportions of altered pixels. Finally, note that the FPRs increased significantly for larger tampered regions involving smaller proportions of tampered pixels. This trend is not surprising, however, as none of the tested methods were aimed at accurately localising altered pixels scattered across the image, e.g. salt-and-pepper noise.

Figure 5.14(a) summarises the self-recovery results gathered when the proportion of altered pixels was 0.55. Zhang and Wang's method exhibited a significantly better



(a)

(b)

(c)

Figure 5.13: Localisation results for the tested proportions of altered pixels: (a) 0.55, (b) 0.75, and (c) 0.95.

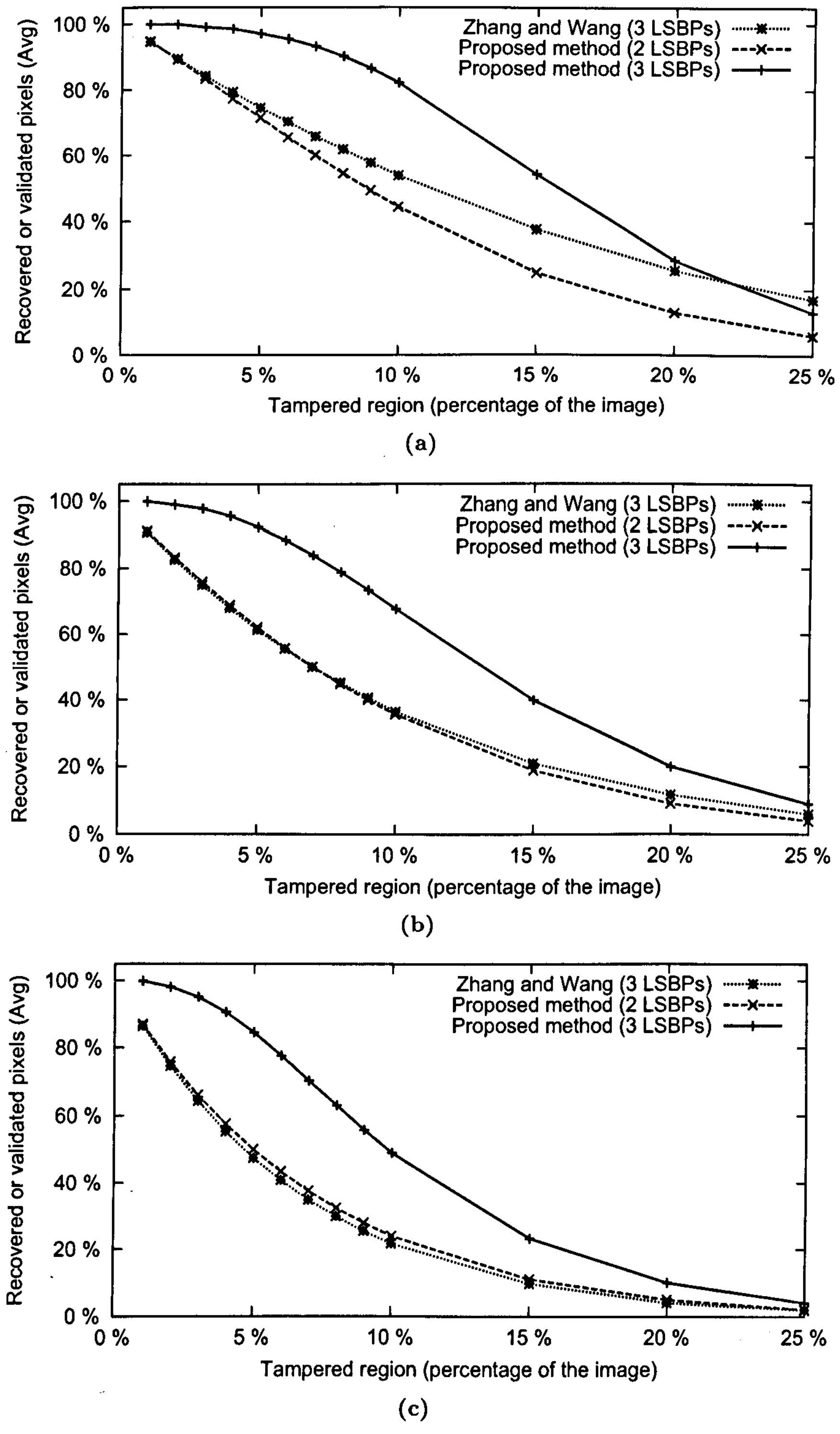


Figure 5.14: Self-recovery results for the tested proportions of altered pixels: (a) 0.55, (b) 0.75, and (c) 0.95.

performance, compared with the proposed scheme using 2 LSBPs. Nonetheless, when using 3 LSBPs for watermarking, the same as in Zhang and Wang's scheme, the proposed method managed to recover/validate a significantly higher percentage of pixels, especially when the tampered region extended to less than 20% of the images. When the tampered region covered 25% of the image, the proposed method was slightly outperformed by Zhang and Wang's scheme. This can be thought of as a side effect of the proposed threshold-based recovery mechanism, which is aimed to skip the analysis of arrays of pixels with many pixels marked as tampered by the block-wise method. However, this is a cheap price to pay for a significantly higher recovery performance in the rest of the tested cases. Figures 5.14(b) and 5.14(c) summarise the self-recovery results for the rest of the tested proportions of altered pixels, namely 0.75 and 0.95, respectively. Observe that, across these tests, there is only a marginal difference between the performance achieved with the proposed method, using 2 LSBPs, and the one achieved with Zhang and Wang's method. However, a significantly higher percentage of altered pixels could be recovered/validated with the proposed scheme when using 3 LSBPs.

5.5 Summary and Conclusions

This chapter has presented a scheme, motivated by Zhang and Wang's idea [65] of combining a pixel-wise and a block-wise mechanisms to afford higher localisation and self-recovery capabilities. Nonetheless, unlike Zhang and Wang's method, the proposed pixel-wise scheme manages to recover pixels associated to subsets containing pixels mistakenly deemed fake by the block-wise method. Furthermore, the parameters of the system can be readily adjusted to control the embedding distortion, as well as the probability of false validations and false recoveries. The pixel-wise method can be iteratively repeated, so that a different bit-plane is watermarked in each iteration. Further still, a block-wise method resilient to cropping has been presented. The information encoded in the watermark is used to restore the original dimensions of the host image and rectify any possible displacement of the content caused by cropping. This way, the synchronisation between the watermark and the pixel-wise method is not affected, thus preserving the localisation enhancement and recovery capabilities of the system.

A set of experiments have been presented to compare the localisation/recovery performance achieved with the proposed method and Zhang and Wang's scheme, in presence of conventional distortions and cropping. Results showed that the proposed method outperforms Zhang and Wang's scheme, especially when the watermark is embedded in the 3 LSBPs of the image (the same as in Zhang and Wang's method). In fact, an example image forgery was generated to evaluate both methods against cropping. Roughly 2.5% of pixels in the forgery had been altered by conventional tampering and 5% of the pixels removed by cropping. Results show that, when using 3 LSBPs, the proposed method managed to recover/validate effectively 6, out of every 10 altered/missing pixels, whereas Zhang and Wang's scheme failed to recover/validate a single pixel.

Additionally, an extensive investigation, conducted on 900 images, has been presented into how the size of the tampered region and the proportion of altered pixels affect the performance of the proposed and Zhang and Wang's method. Results show that the localisation performance achieved with the proposed scheme, using 2 LSBPs, is comparable to the performance exhibited by Zhang and Wang's method. Nonetheless, these were outperformed by the proposed method when using 3 LSBPs, especially in tests involving greater proportions of altered pixels. Results showed that the localisation performance of the two methods tend to decay in forgeries involving larger tampered regions with smaller proportions of altered pixels. Therefore, neither the proposed scheme nor Zhang and Wang's method are suitable to recover images containing scattered changes, such as salt-and-pepper noise. The method in [67] can be used to recover images affected by this type of distortion, as long as the number of tampered pixels is less that 3.2% of the image. Self-recovery results showed that, when using 2 LSBPs, the performance achieved by proposed method and Zhang and Wang's scheme (which uses 3 LSBPs) is comparable in forgeries involving greater proportions of altered pixels. Nevertheless, a much better performance was exhibited by the proposed method when using 3 LSBPs.

Chapter 6

Automated Detection of Duplicated Regions Affected by Reflection, Rotation and Scaling

In this chapter, an automated mechanism is proposed to detect duplicated regions in a forensic fashion, without relying in watermarks or digital signatures. The core mechanism uses colour-based features and 1-D descriptors derived from log-polar maps to perform efficiently in terms of memory usage and computational cost.

In Section 3.3.1, it has been shown that, even though the detection of duplicated regions plays an important role in image forensics, most of the existing methods are too sensitive to geometric changes in the replicated areas. Although this concern has been addressed in [84–90], none of these methods copes with duplicates affected by reflection, rotation and scaling in a comprehensive manner.

Forensic techniques designed for detecting image traces of interpolation can be used to identify regions affected by rotation and scaling [91–94, 117]. However, duplicated areas affected by geometric distortions that do not require re-sampling (e.g. reflection) would go unnoticed by these methods. Furthermore, the blocking artifacts introduced by JPEG compression, significantly affect the detection performance of these schemes.

To overcome all these shortcomings, a forensic method is proposed to detect duplicated regions that have undergone reflection, rotation or/and scaling. Overlapping blocks of pixels are independently mapped to 1-D reflection/rotation-invariant descriptors derived from log-polar maps. This dimension-reduced representation of the blocks enables an efficient search for duplicates in terms of memory space. In addition, a double-check refinement mechanism is proposed to identify clusters of duplicated blocks that have been affected by geometric distortions. Extensive experimental results demonstrate the effectiveness of the proposed method.

The remainder of this chapter is organised as follows. The proposed 1-D descriptors and the forensic method are presented in Sections 6.1 and 6.2, respectively. Section 6.3 elaborates on the parameter settings, while Section 6.4 presents extensive experimental results, along with a comparative evaluation of the proposed scheme against Myna *et al.*'s approach [83]. Finally, a summary of the chapter is presented in Section 6.5, along with some conclusions.

6.1 Solving Reflection, Rotation and Scaling

To cope with reflection, rotation and scaling, blocks of pixels are mapped to 1-D descriptors, resulting from summing the pixels in their log-polar representation along the angle axis. This approach was motivated by the watermarking technique presented in [138], which was designed to overcome the synchronisation problems caused by rotation, scaling and translation (reflection was not considered in this method). In that case, 1-D descriptors were computed by summing a log-polar map along the log-radius axis, thus achieving properties distinct to the descriptors proposed below.

6.1.1 Computing the 1-D descriptors

Consider the point $(x, y) \in \mathbb{R}^2$, which can be written using (natural) log-polar coordinates,

$$x = \exp(\rho)\cos\theta, \text{ and, } y = \exp(\rho)\sin\theta \ , \tag{6.1}$$

where $\rho \in \mathbb{R}$ and $0 \leq \theta < 2\pi$. Let (x', y') denote the coordinates of a reflected, rotated and scaled point, i.e.,

$$\begin{aligned} x' &= \mu(x\cos\varphi + y\sin\varphi) \ , \\ y' &= \mu(x\sin\varphi - y\cos\varphi) \ , \end{aligned} \tag{6.2}$$

where φ and μ are the parameters of rotation and scaling, respectively. Rewriting Equation (6.2) in log-polar form,

$$\begin{aligned} x' &= \exp^{(\rho+\log\mu)}\cos(\varphi - \theta) \ , \\ y' &= \exp^{(\rho+\log\mu)}\sin(\varphi - \theta) \ . \end{aligned} \tag{6.3}$$

Observe that scaling in rectangular coordinates results in a simple translation of the log-polar map. Consider a block of pixels[1] $B_i(x, y)$ and its log-polar representation $B_i(\rho, \theta)$. A 1-D descriptor $\vec{v}_i$ can be defined as,

$$\vec{v}_i(\rho) = \sum_{\theta} B_i(\rho, \theta) \ . \tag{6.4}$$

A reflected, rotated and scaled version of B_i can be expressed in log-polar coordinates as $B_i'(\rho, \theta) = B_i(\rho + \log\mu, \varphi - \theta)$. A descriptor for this block can be calculated

[1] The index i is merely included for the sake of consistency with the notation employed in the next section.

as follows,

$$\begin{aligned}\vec{v}_i{}'(\rho) &= \sum_{\theta} B_i'(\rho,\theta) \\ &= \sum_{\hat{\theta}} B_i(\rho + \log\mu, \hat{\theta}) \ ,\end{aligned} \tag{6.5}$$

where $\hat{\theta} = (\varphi - \theta)$. Since cosines and sines are periodic functions, and a whole period is being considered, and the change in $\hat{\theta}$ is linear, it follows that Equation (6.5) can be rewritten as,

$$\vec{v}_i{}'(\rho) = \sum_{\theta} B_i(\rho + \log\mu, \theta) \ . \tag{6.6}$$

Hence, the descriptors are invariant to both reflection and rotation, i.e. $\vec{v}_i(\rho) = \vec{v}_i{}'(\rho)$, when B_i' is a mirrored and/or rotated version of B_i. Furthermore, $\vec{v}_i(\rho) = \vec{v}_i{}'(\rho - \log\mu)$, when B_i' is a scaled version of B_i by a factor μ. In discrete signals though, loss due to interpolation and round-off errors has to be considered. Because of the well-known translation properties of the Fourier transform [139], the Fourier magnitude of both descriptors is expected to be very closely correlated to each other, i.e.,

$$c(\vec{V}_i, \vec{V}_i') = \frac{\vec{V}_i^T \vec{V}_i'}{\sqrt{(\vec{V}_i^T \vec{V}_i)(\vec{V}_i'^T \vec{V}_i')}} \approx 1 \ , \tag{6.7}$$

where c is the correlation coefficient, $\vec{V}_i$ and $\vec{V}_i'$ are the Fourier magnitudes of $\vec{v}_i$ and $\vec{v}_i{}'$, respectively.

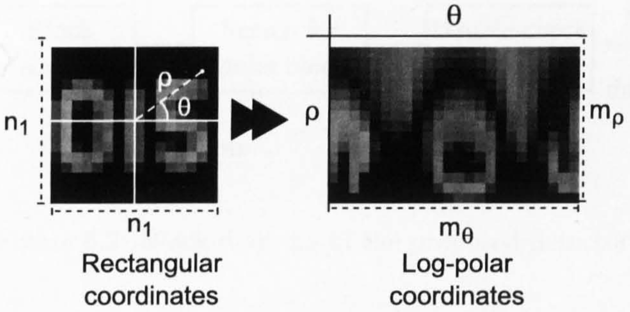


Figure 6.1: Illustration of log-polar mapping.

6.1.2 Practical considerations

As stated above, the 1-D descriptors result from the sum of the pixels of a log-polar map along the angle axis. The transformation from rectangular coordinates to log-polar coordinates, illustrated in Figure 6.1, is typically referred to as log-polar mapping (LPM). To compute this transformation, the origin of the rectangular coordinates is assumed to be the centre of the image, as in Figure 6.1 (left). Because of the discrete nature of digital images, the parameters of their log-polar representation are also discrete, and depend on the size of the of the log-polar map m_ρ and m_θ. In the present investigation, it was empirically found that, in general, 1-D descriptors derived from log-polar maps sized $n_1 \times \lfloor \pi\, n_1/2 \rfloor$ provide sufficient resilience to rotation, scaling and JPEG compression. For example, the dimensions of the log-polar representation of a block of 24×24 pixels should be $24 \times \lfloor \pi\, 24/2 \rfloor = 24 \times 37$; that is $m_\rho = 24$ and $m_\theta = 37$. Nonetheless, a more exhaustive investigation may be useful to improve their performance.

6.1.3 Advantages of 1-D descriptors

Compared to complete log-polar maps, the main benefits of 1-D descriptors are the following:

1. *Reflection* - As shown above, the detection of reflected duplicated blocks is drastically simplified.

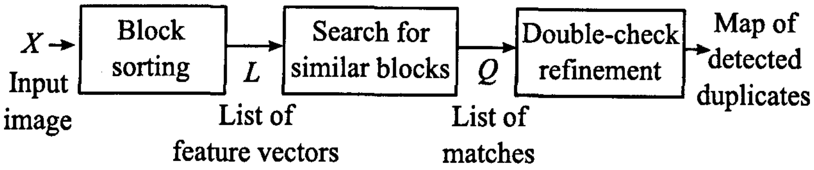


Figure 6.2: Block-diagram of the proposed detector.

2. *Memory* - Since every single block will be compared with several blocks, it is a good idea to form a list of descriptors, each one corresponding to a different block, before the search stage. Thus, instead of storing a complete $m_\rho \times m_\theta$ log-polar map for every block, it will only be necessary to store a vector of length m_ρ, thereby reducing substantially the memory usage.

3. *Computational cost* - The calculation of the proposed descriptors involves exactly the same number of operations as the complete log-polar map. Nonetheless, the computational cost of the whole system decreases, as it will only need to compute the fast Fourier transform (FFT) of every 1-D descriptor, whose complexity is given by $O(m_\rho \log_2 m_\rho)$, instead of computing the FFT of complete log-polar maps, whose complexity is $O((m_\rho m_\theta)\log_2(m_\rho m_\theta))$.

6.2 Proposed Algorithm

Consider a colour image X, of size $n_1 \times n_2$, which is tiled as blocks of pixels selected by sliding, pixel by pixel, a window of size $q \times q$, from the top-left corner to the bottom-right corner, in a raster-scan order. Let A_i denote the i-th block of pixels, for $i = 1, \ldots, (n_1 - q + 1)(n_2 - q + 1)$. The proposed method is comprised of the three general stages shown in Figure 6.2.

6.2.1 Block sorting

The blocks are sorted to reduce the computational cost of the search stage. The centre of each block A_i will be the centre of a disc of diameter q, which fits just inside the block. Consider the three features f_{1_i}, f_{2_i} and f_{3_i}, which denote the average of the red, blue and green colour components, respectively, of the pixels within the disc. Luo *et al.* [79] pointed out that the average of the colour channels is not significantly modified by JPEG compression or Gaussian blurring. Section 6.3 will show that such colour-dependent features are resilient to rotation and mild scaling.

A problem typically faced by methods aimed at detecting duplicates is caused by areas of uniform luminance (e.g. sky) [5], which can produce a significant number of false duplicates. Furthermore, these areas can increase the computational cost, as overlapping blocks in such regions are very similar. To overcome this, a fourth feature is proposed to discard blocks of uniform luminance values. The luminance of each pixel within a disc is calculated as $\mathrm{Y} = 0.2126\,\tilde{r} + 0.7152\,\tilde{g} + 0.0722\,\tilde{b}$ [101], where $\tilde{r}$, $\tilde{g}$ and $\tilde{b}$ are components of red, green and blue, respectively. The probability distribution function is calculated using the luminance of all the pixels within a disc. From this, the entropy is calculated as, $f_{4_i} = -\sum_k p_k \log_2 p_k$, where p_k is the probability of each luminance value in the disc. Since blocks with uniform luminance generate low entropy values, blocks whose entropy falls below a predefined threshold $e_{\min}$ will be discarded. Then a list L is formed with the feature vectors $(f_{1_i}, f_{2_i}, f_{3_i}, f_{4_i})$, derived from the remaining blocks, and then sorted, in ascending order. This sorting order is typically referred to as *lexicographical order* [5, 76].

6.2.2 Search for similar blocks

Let B_i be the i-th block of luminance values corresponding to the i-th feature vector in L. Additionally, let (x_i, y_i) be the coordinates (the centre) of B_i in the image X.

A descriptor $\vec{v}_i$ is computed for every B_i by Equation (6.4) and its Fourier magnitude $\vec{V}_i$ is calculated. Duplicated blocks are expected to have produced similar feature vectors, even if they have undergone reflection, rotation and/or scaling (see Section 6.3). Hence, the search for duplicates can be drastically reduced, as only blocks with similar features will be compared. Thus, the correlation coefficient $c_{ij} \equiv c(\vec{V}_i, \vec{V}_j)$ is computed, for every $j > i$ that satisfies the following conditions:

a) $d_{ij} > \tau_d$,

b) $|f_{k_i} - f_{k_j}| \leq \tau_h$, for $k = 1, 2, 3$, and

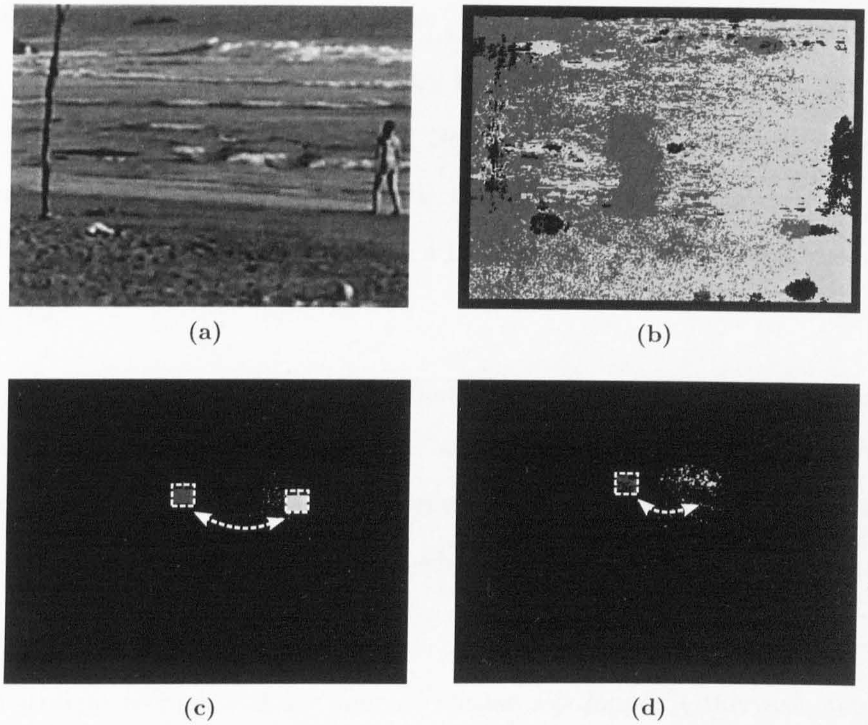
c) $|f_{4_i} - f_{4_j}| \leq \tau_e$,

where τ_d, τ_h and τ_e are predefined thresholds, discussed in Section 6.3, and $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. The first condition is necessary to avoid comparing blocks in proximity to each other, whereas the two other conditions are used to determine whether two blocks are similar enough to be considered potential duplicates. Note that, since L is sorted, the comparisons for $\vec{V_i}$ can stop once a descriptor $\vec{V_u}$ is reached, such that $f_{1_u} > (f_{1_i} + \tau_h)$. Let c_{ir} be the higher correlation coefficient computed for $\vec{V_i}$. If c_{ir} lays above a predefined similarity threshold τ_{sim}, calculate the offsets defined as $\delta^x_{ir} = |x_i - x_r|$ and $\delta^y_{ir} = |y_i - y_r|$. Then, if $x_i <= x_r$, generate a tuple $(\delta^x_{ir}, \delta^y_{ir}, x_i, y_i, x_r, y_r)$; otherwise, create a tuple $(\delta^x_{ir}, \delta^y_{ir}, x_r, y_r, x_i, y_i)$. For convenience, the first and second pairs of coordinates in a tuple will be referred to as "source" and "target" coordinates, respectively. Such a layout of coordinates are essential for the next refinement stage. To complete the current stage, append all the tuples to a lexicographically sorted list Q.

6.2.3 Double-check refinement

At this point, the list Q is expected to contain a huge number of false pairs. To illustrate the importance of the refinement stage, the unrefined matches found in the forgery in Figure 6.3(a) are shown in Figure 6.3(b); the "source" and the "target" coordinates are depicted by dark-grey and light-grey spots, respectively. Observe that the detector response would be useless without an additional refinement stage.

The aim of this stage is to discriminate possible *valid duplicates*, from false matches. When a region has been cloned without further geometric distortions, the detection of


(a)

(b)

(c)

(d)

Figure 6.3: Example unrefined detection results. (a) Example forgery submitted to the detector. (b) Unrefined pairs detected in the example forgery; points given by "source" coordinates (dark-grey) and "target" coordinates (light-grey) (c) Example of valid duplicate. (d) Example of invalid duplicate.

valid duplicates can be confined to the search for densely packed groups of tuples with identical offsets [5, 76, 79]. However, this approach is utmost unsuitable when the replicated areas have undergone geometric distortions. Hence, an iterative double-check refinement mechanism is proposed to efficiently identify clusters of points formed by tuples, in Q, containing similar –i.e. not necessarily identical– offsets[2].

Let Δ be a user-defined offset employed in the refinement process; in practice, it was found that the system performs well with $\Delta = 32$. The proposed steps are described below:

1. In this step, a new list is generated with the coordinates of the tuples containing

[2] Among various tested strategies, the presented double-check refinement mechanism is the one that led to better results.

offsets nearby $\delta^x_{a_1b_1}$ and $\delta^y_{a_1b_1}$. Let $(\delta^x_{a_1b_1}, \delta^y_{a_1b_1}, x_{a_1}, y_{a_1}, x_{b_1}, y_{b_1})$ be the front-tuple in Q. Create a tuple $(x_{a_j}, y_{a_j}, x_{b_j}, y_{b_j})$ for every j, such that $\delta^x_{a_jb_j} \leq (\delta^x_{a_1b_1} + 2\Delta)$, and $\delta^y_{a_1b_1} \leq \delta^y_{a_jb_j} \leq (\delta^y_{a_1b_1} + \Delta)$. Note that, since Q is sorted, the search is completed once a descriptor $\delta^x_{a_ub_u}$ is reached, such that $\delta^x_{a_ub_u} > (\delta^x_{a_1b_1} + 2\Delta)$. All the new tuples are appended to a list F, sorted by the "source" coordinates, x_{a_j}, y_{a_j}.

2. Considering only the "source" coordinates, search F for clusters with more than t_{min} points within a window of size[3] $w \times w$; such clusters will be referred to as "source" clusters. A new list is generated with the tuples, in F, that form each "source" cluster (a different list for each cluster). Let G_k denote the list associated to the k-th detected cluster.

3. Go straight to Step 4 if no "source" cluster was found. Otherwise, considering only the "target" coordinates, every G_k is independently searched for clusters with more than t_{min} points within a $w \times w$ window; such clusters will be regarded as to "target" clusters. Tuples that simultaneously form both a "source" cluster and a "target" cluster are deemed valid duplicates, and thus are appended to a set S. An example of a valid duplicate is illustrated in Figure 6.3(c); the $w \times w$ window is depicted as a dashed white square. This double-check mechanism allows us to readily discard invalid duplicates, like the one shown in Figure 6.3(d), where the "target" coordinates associated with the identified "source" cluster are noticeably scattered; i.e. no "target" cluster was formed.

4. For the sake of memory efficiency, some of the tuples that have been analysed already will be deleted from Q. In particular, remove every tuple j, such that $_x\delta_{a_jb_j} \leq (\delta^x_{a_1b_1} + 2\Delta_{\text{step}})$, and $\delta^y_{a_1b_1} - \Delta_{\text{step}} \leq \delta^y_{a_jb_j} \leq (\delta^y_{a_1b_1} + \Delta_{\text{step}})$, where Δ_{step} is a user-defined value, such that $\Delta_{\text{step}} < \Delta$. It has been empirically observed

[3] It was observed that the values $w = 16$ and $t_{\text{min}} = 230$ led to the better results when 24×24 blocks are employed in the Block sorting stage (recall Section 6.2.1). Nonetheless, t_{min} should be slightly increased for larger block-sizes.

that $\Delta_{\text{step}} = 16$ keeps a fair balance between computational cost and detection accuracy. Next, go to Step 1 if Q is not empty.

Finally, the tuples in S are used to encode a bitmap to illustrate the detected duplicates.

6.3 Parameters Set Up

To find suitable values for the thresholds τ_h, τ_e and τ_{sim}, required in Section 6.2.2, the following experiment was conducted using 700 images, sized 640 × 480, in the Caltech-256 dataset [137]. The test images depicted diverse contents, from objects to landscapes, excluding synthetically generated content (e.g. vector graphics). Such diversity is expected to allow us to define threshold values suitable for the analysis of a vast variety of images. First, the images were split into non-overlapping pixel blocks. Two block-sizes, 24 × 24 and 32 × 32, were individually tested. The 35 distortions listed below were tested, one at a time, in each pixel block.

- *JPEG compression* - The tested JPEG-quality factors varied from 60 to 100 with increments of 10.
- *Rotation* - The tested rotation degrees[4] varied from 0° to 90° with increments of 5°.
- *Scaling*- The tested scaling factors[4] varied from 0.95 to 1.05 with increments of 0.01.

It was observed that the absolute difference between the colour-dependent features (f_1, f_2 and f_3) extracted from the distorted blocks and from the original blocks remained below 2 in over 93% of the total cases. Additionally, in over 97% of the cases, the absolute difference between the entropy-dependent feature (f_4) computed from the

[4] The distorted blocks were additionally translated a random amount within the range $[-0.5, 0.5]$, in a random direction, to emulate real forgeries.

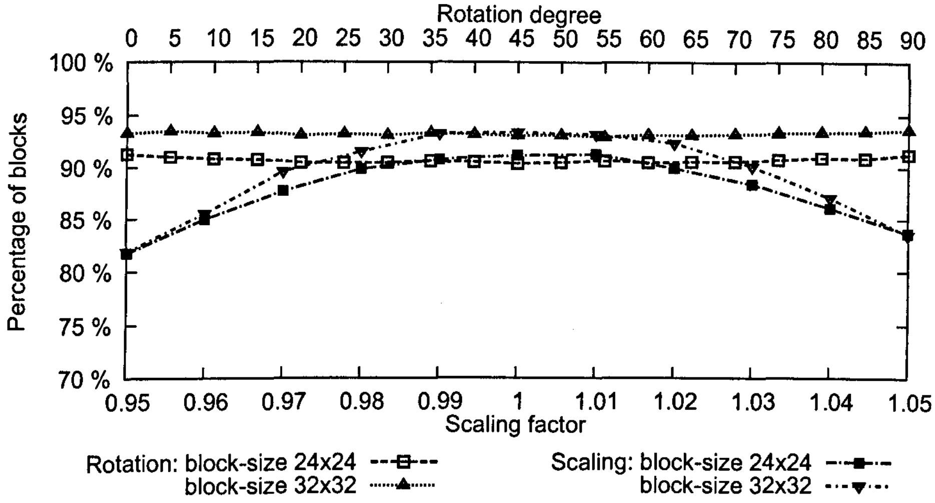


Figure 6.4: Percentage of blocks that satisfied the predefined thresholds.

distorted blocks and from the original blocks remained below 0.3. On the other hand, the correlation coefficient between the Fourier magnitude of the 1-D descriptor extracted from the distorted blocks and from the original blocks laid above 0.9995 in over 90% of the total cases. Hence, the thresholds were set as: t_h = 2, t_e = 0.3and $t_{\text{sim}} = 0.9995$.

Figure 6.4 shows the percentages of blocks whose features simultaneously satisfied the conditions b) and c), in Section 6.2.2, using the thresholds above. For rotation, results are very similar independent of the rotation degree for each of the block-sizes of 24 × 24 and 32 × 32. Yet, results for 32 × 32 blocks are better than those for 24 × 24 blocks. For scaling, results increase as the scaling factor tends to 1 (for either end) for each of the block-sizes of 24 × 24 and 32 × 32. Although results for both block-sizes are virtually the same at scaling factor values of 0.95 and 1.05, those for 32 × 32 blocks tend to be better than those for 24 × 24 blocks, as the scaling factor approaches 1. For the sake of brevity, only the results for scaling and rotation are presented. However, it is worth mentioning that over 97% of the blocks satisfied the thresholds, even after compression using JPEG-quality factor of 60.

6.4 Experimental Results

Various experiments were conducted to evaluate the detection performance of the proposed system. The block-sizes 24 × 24 and 32 × 32 were tested using the parameter settings discussed in the preceding section; additional parameters were: $e_{\min} = 2$ and $\tau_d = 40$. Furthermore, in Section 6.4.3, a detailed comparison between the proposed method and Myna *et al.*'s scheme [83] is presented. The interest in Myna *et al.*'s scheme stems from the fact that, unlike Lin *et al.*'s method [84], it is aimed at detecting duplicates that have undergone scaling and rotation by arbitrary angles[5].

The two standard measures, TPR and TNR, defined in Equation (4.19) will be adopted to evaluate the localisation performance quantitatively. Recall that an ideal detection would simultaneously render TPR=1 and TNR=1.

6.4.1 Example forgeries

To begin with, four example forgeries were created to illustrate the detection capabilities of the proposed method:

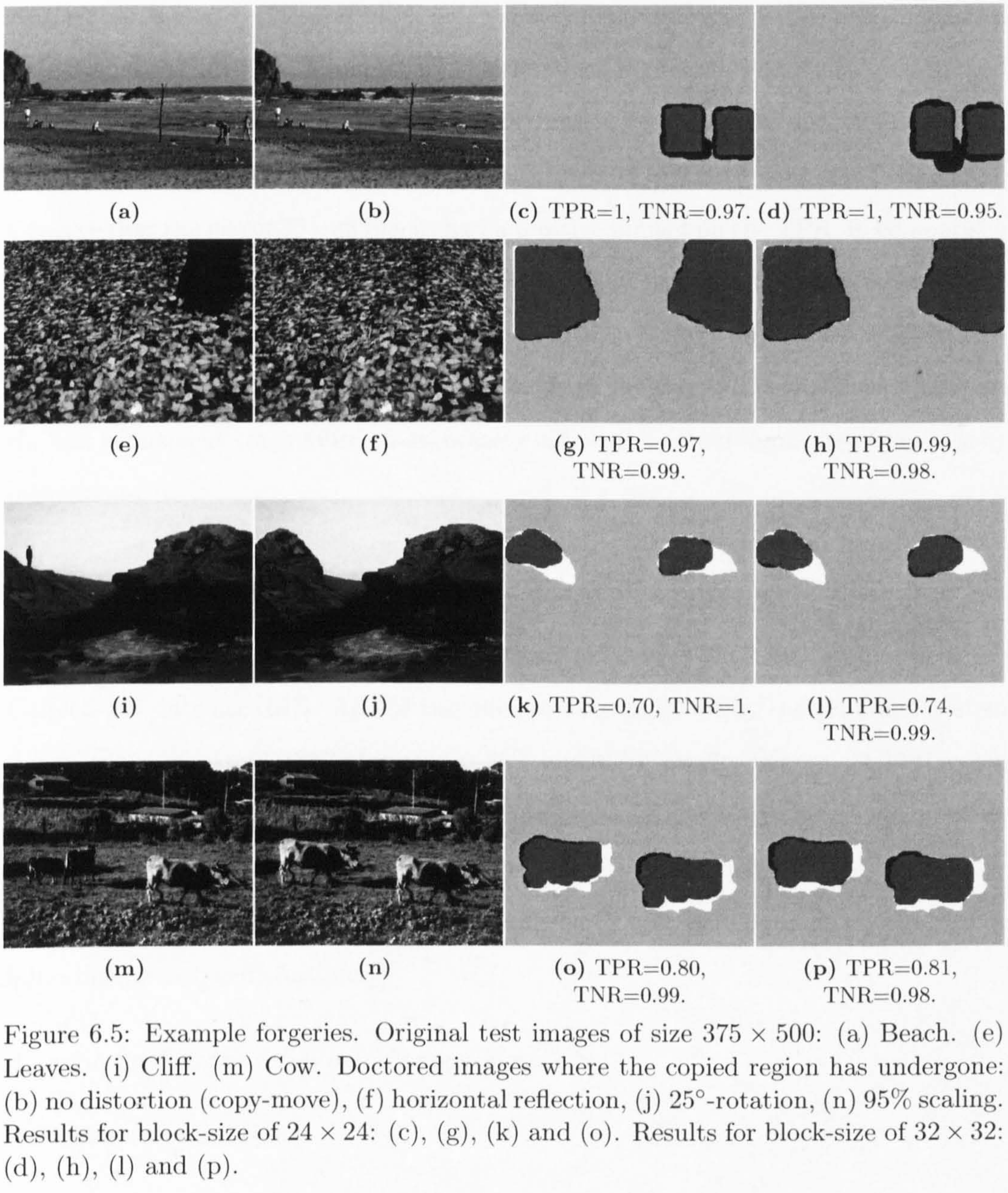
- The girls in proximity to the lower-right corner in Figure 6.5(a) were overlaid with a region copied from a nearby location to produce Figure 6.5(b).

- A region by the upper-left corner, in Figure 6.5(e), was horizontally flipped and used to conceal the treeand generate Figure 6.5(f).

- The person on the cliff, in Figure 6.5(i), was covered by a 25°-rotated portion of the rocks in the opposite side to get the forgery in Figure 6.5(j).

- A copy of the cow, in Figure 6.5(m), was scaled 95% of its original size and used to conceal the rest of the animals in the scene. The produced forgery is shown in Figure 6.5(n).

[5] Comparing with Huang *et al.*'s method [87] was rather impractical. The results from this method include a set of key points, but the duplicated regions are not delimited.



(a)

(b)

(c) TPR=1, TNR=0.97.

(d) TPR=1, TNR=0.95.

(e)

(f)

(g) TPR=0.97, TNR=0.99.

(h) TPR=0.99, TNR=0.98.

(i)

(j)

(k) TPR=0.70, TNR=1.

(l) TPR=0.74, TNR=0.99.

(m)

(n)

(o) TPR=0.80, TNR=0.99.

(p) TPR=0.81, TNR=0.98.

Figure 6.5: Example forgeries. Original test images of size 375 × 500: (a) Beach. (e) Leaves. (i) Cliff. (m) Cow. Doctored images where the copied region has undergone: (b) no distortion (copy-move), (f) horizontal reflection, (j) 25°-rotation, (n) 95% scaling. Results for block-size of 24 × 24: (c), (g), (k) and (o). Results for block-size of 32 × 32: (d), (h), (l) and (p).

The bitmaps encoded by the detector, using 24×24 blocks, were compared with their respective ground truth as shown in Figures 6.5(c), 6.5(g), 6.5(k) and 6.5(o). Darker grey areas depict duplicated regions successfully detected, white areas represent unnoticed duplicated regionsand black areas show regions mistakenly deemed duplicates. The results using 32 × 32 blocks are presented in Figures 6.5(d), 6.5(h), 6.5(l) and 6.5(p). From this small set of 4 images, experiments from different operations and different block-sizes indicate typically high average values of TPR (0.88) and TNR (0.98). Observe that the use of 32×32 blocks had a positive impact on the TPR, in comparison with the results obtained using 24 × 24 blocks, at the expense of slightly affecting the TNR. Moreover, note that the best values of TPR and the worst values of TNR correspond to the same test (the simple copy-move in the Beach image). This is because the last refinement stage failed to accurately isolate the actual duplicates from nearby mismatches.

6.4.2 Extensive tests

This experiment was conducted on 100 test images, sized 400 × 600, available in the Caltech-256 data set [137]. All the test images were analysed by the proposed system in non-compressed and JPEG format (at 80% and 100% quality factors). Additionally, every image was used to generate a set of 36 forgeries as follows. An 80 × 80 pixel block was selected from a random location in the image. Before being pasted on another random location within the same image, the block was manipulated by one of the following geometric distortions:

- No further distortion (simple copy-move).
- Horizontal reflection
- Rotation: 5°, 20°, 40°, 60°80°, 100°, 120°, 140°, 160°175° or 180°.
- Scaling factors: 0.96, 0.98, 1.01, 1.03 or 1.05.

Table 6.1: Results of false detections, out of a total 100 original images.

Detector's block-size	non-compressed format	JPEG quality factor 100%	 80%
24 × 24	33	32	33
32 × 32	20	19	21

The same procedure was repeated using 120 × 120 duplicates. Each forgery was analysed in non-compressed and JPEG format (at 100% and 80% quality factors). Hence, a total of 10,800 manipulated images were analysed. On average, the proposed system produced an answer for an image in less than 14 minutes, when using 24 × 24 blocks, and in less than 15 minutes, when using 32 × 32 blocks.

Table 6.1 presents the number original images that were mistakenly regarded as containing duplicates (TNR $\neq 1$). Observe that the results obtained when using 32×32 blocks compare favourably to those achieved with 24×24 blocks. The tested mild JPEG compression did not have a substantial impact on the number of false detections.

The number of correctly detected forgeries (TPR$\neq$ 0), using 24 × 24 blocks, are summarised in Figures 6.6(a), 6.6(b) and 6.6(c). Results obtained using 32 × 32 blocks are shown in 6.7(a), 6.7(b) and 6.7(c). In general, the detection performance of the proposed system is significantly better when the forgeries contained a larger duplicate. Observe that the detection performance resulting from the two tested block-sizes is comparable when the forgeries contained 120 × 120 duplicates. Nonetheless, when the forgeries contained duplicates of size of 80 × 80, using 24 × 24 blocks led to a slightly higher the number of correct detections, especially in the rotation and scaling tests. Results also show that JPEG compression can slightly affect the detection performance, specially for smaller duplicates.

The localisation results obtained when using a block-size of 24 × 24 are summarised in Figures 6.8(a), 6.8(b), 6.8(c), 6.9(a), 6.9(b) and 6.9(c). The highest average TPR (0.96), calculated from both duplicate-sizes, were obtained from the reflection test, while the 140°-rotation test led to the lower average TPR (0.58). Nonetheless, it was noticed that the tests that led to higher TPRs are the ones that also led to lower TNRsand vice versa. For example, the average TNRs obtained from the reflection and the 140°-rotation tests were assessed to be 0.98 and 0.99, respectively. The localisation performance achieved when using 32×32 blocks was very similar, as observed in Figures 6.10(a), 6.10(b), 6.10(c), 6.11(a), 6.11(b) and 6.11(c). For example, both the higher average TPR (0.96) and the lower average TNR (0.98) were obtained from the reflection test. Additionally, the lower average TPR (0.58) and the higher TNR (0.99) were obtained from the 140°-rotation test.

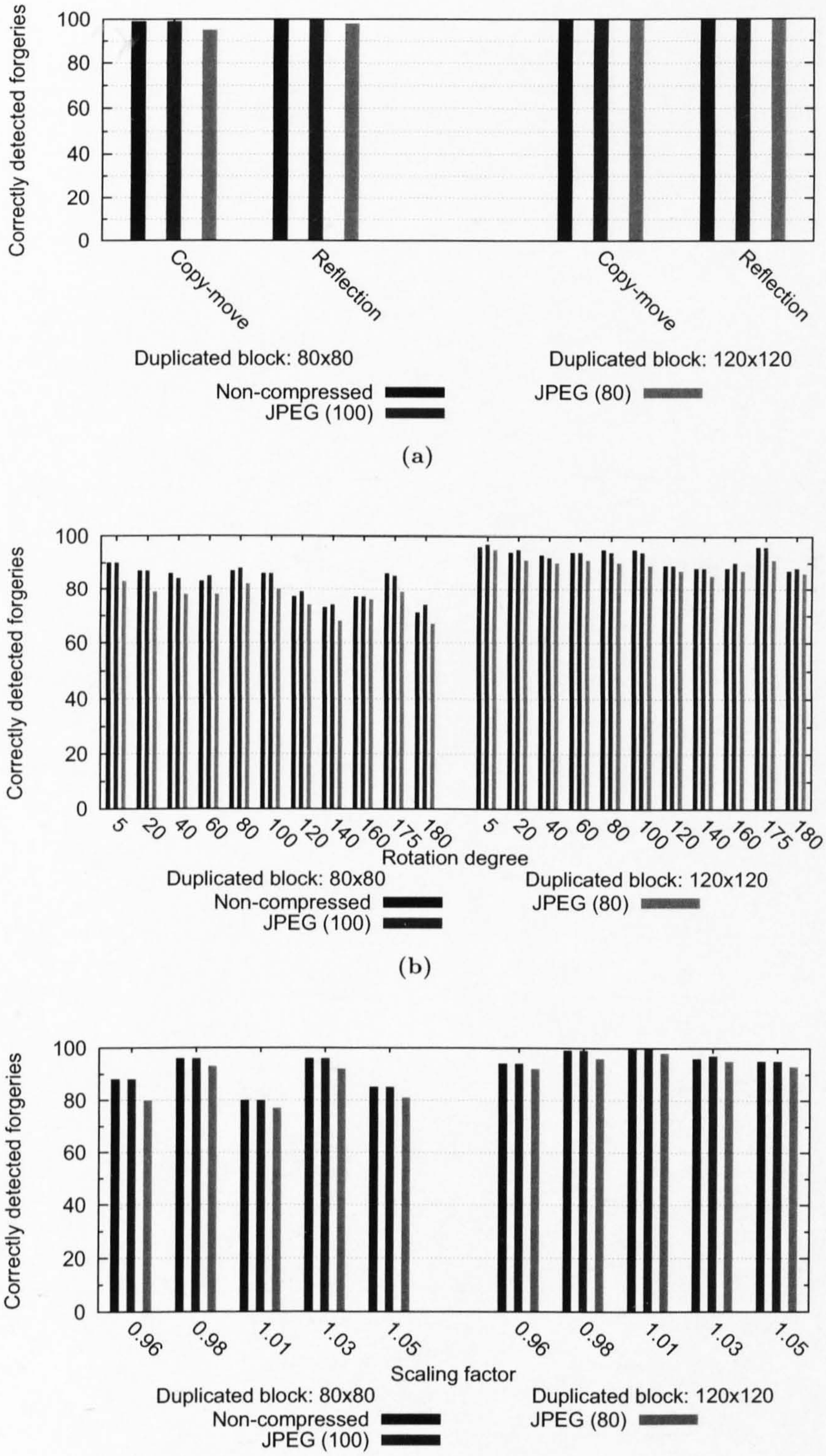
Results also suggest that, the adverse impact that mild JPEG compression has on the localisation performance can be mitigated by using a larger block-size of 32 × 32.

6.4.3 Comparison tests

A total of 20 images sized 300 × 400 were used to compare the proposed method with Myna *et al.*'s scheme[6] [83]. To create a forgery, a block of 100 × 100 pixels was chosen from a random location in the image. Before being pasted on another random location of the same image, the block was subjected to one of the following distortions:

- No further distortion (simple copy-move).
- Horizontal reflection (Ref.).
- Rotation (Rot.*); random rotation angle selected from {90°, 180°, 270°}.
- Rotation (Rot.); random rotation angle within the range (0°, 360°).
- Scaling (Sc.); random scaling factor within the range [0.95, 1.05].
- Rotation and Scaling (Rot.+Sc.); random rotation angle within the range (0°, 360°) and random scaling factor within the range [0.95, 1.05].

[6] As the authors in [83] suggested no suitable settings, every effort has been made to find the parameter set-up that provided the better results in Myna *et al.*'s method.

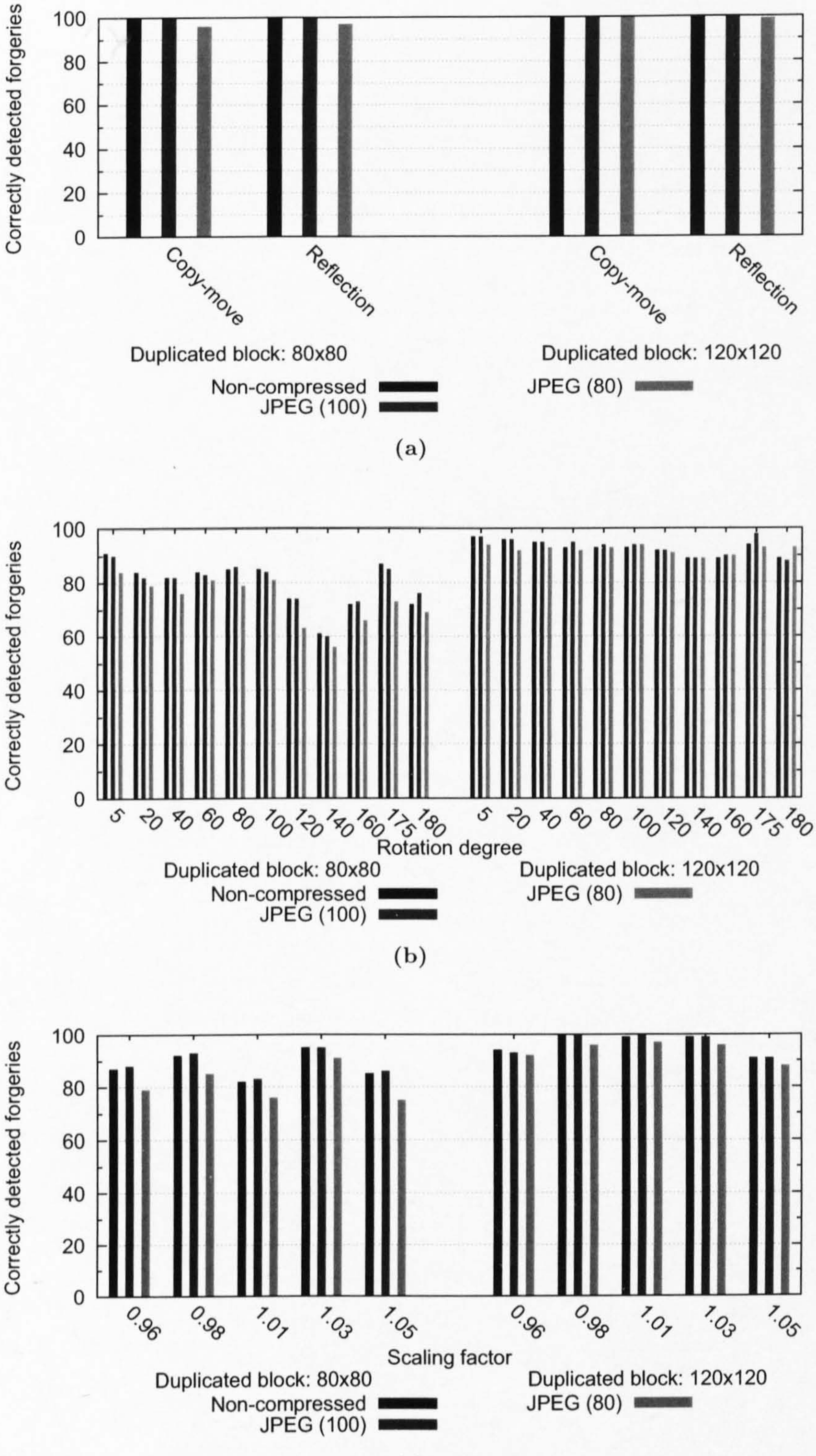


(a)

(b)

(c)

Figure 6.6: Detection results of the extensive tests using a block-size of 24 × 24.

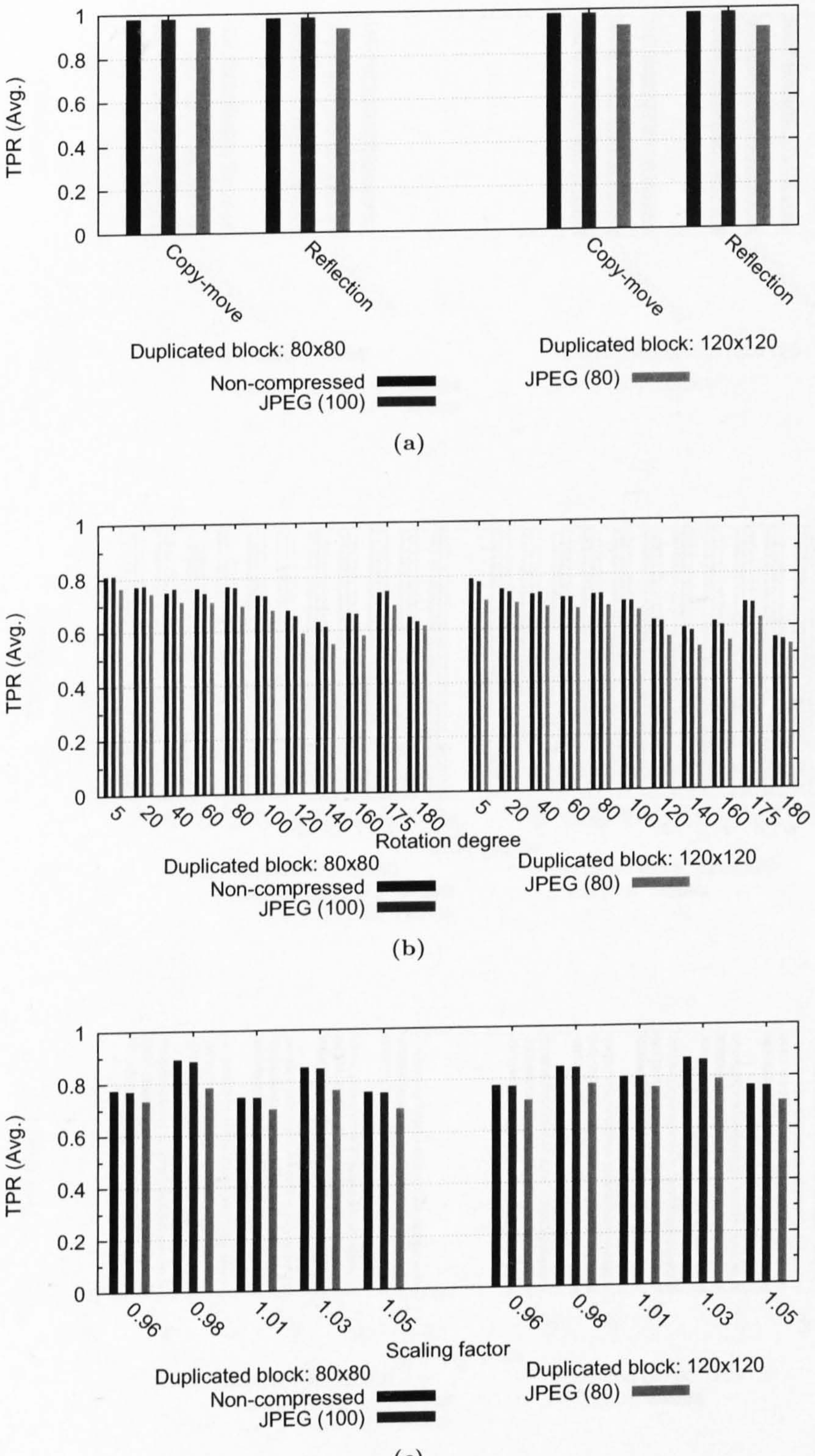


(a)

(b)

(c)

Figure 6.7: Detection results of the extensive tests using a block-size of 32 × 32.

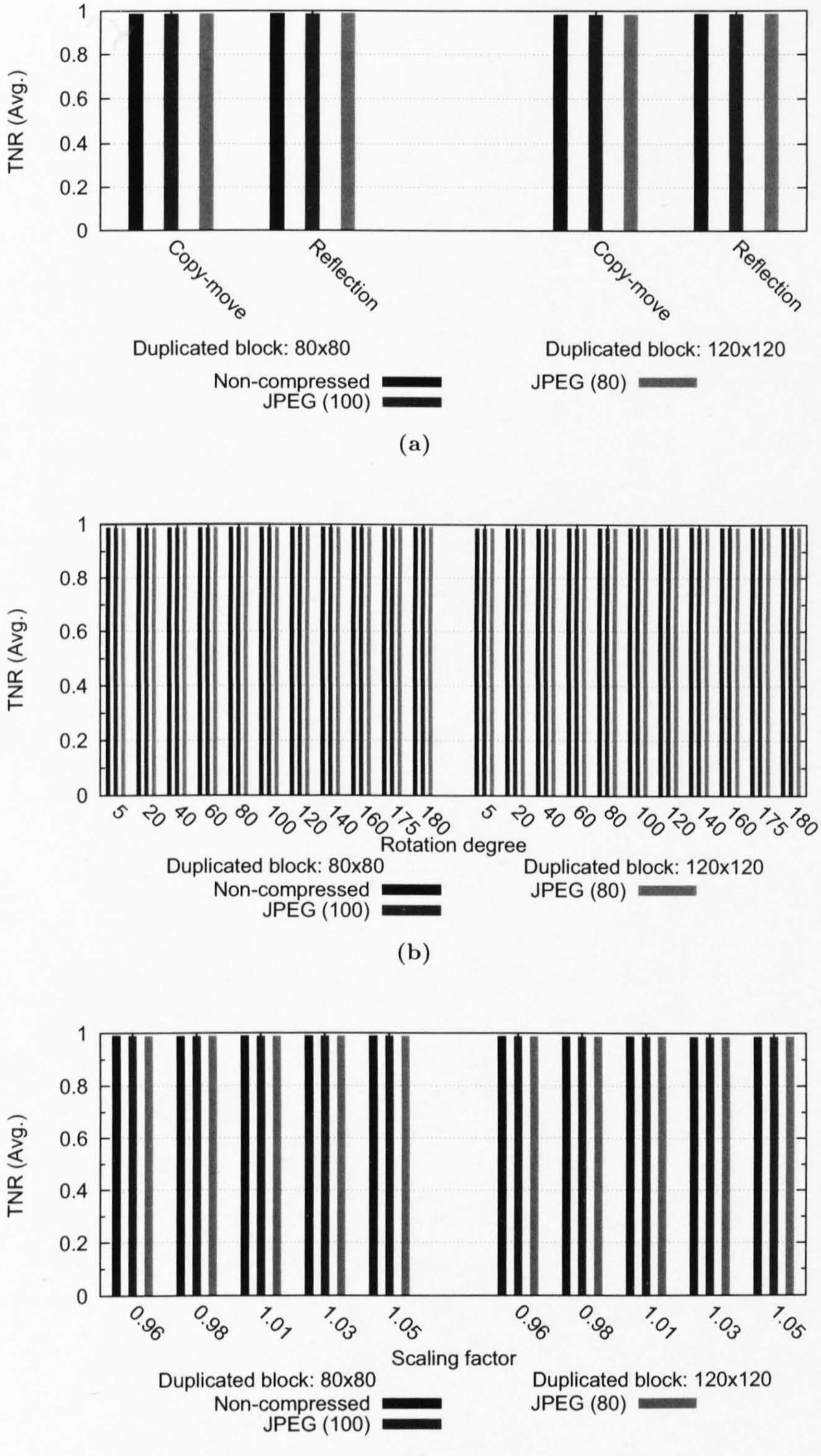


(a)

(b)

(c)

Figure 6.8: TPR results of the extensive tests using a block-size of 24 × 24.

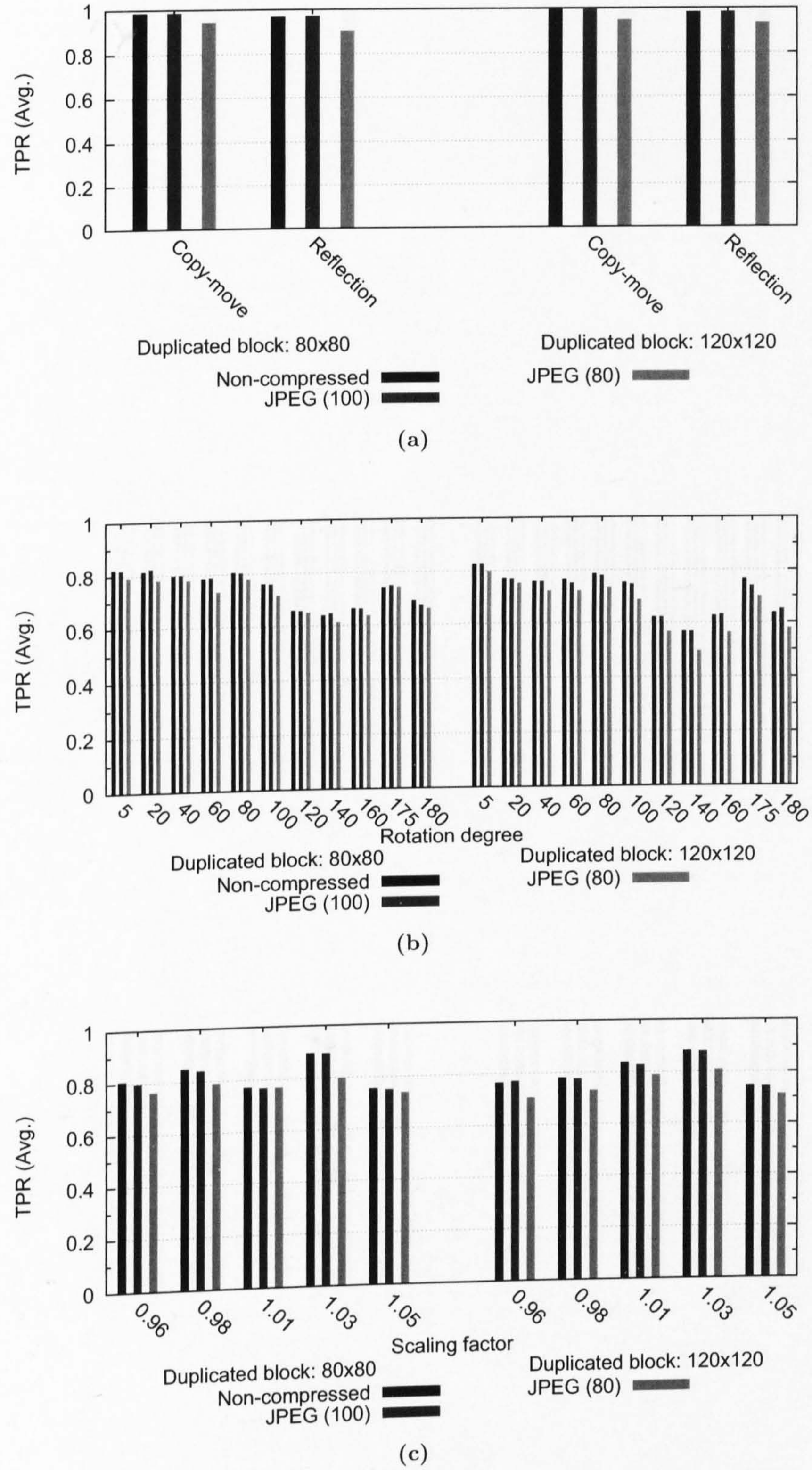


(a)

(b)

(c)

Figure 6.9: TNR results of the extensive tests using a block-size of 24 × 24.

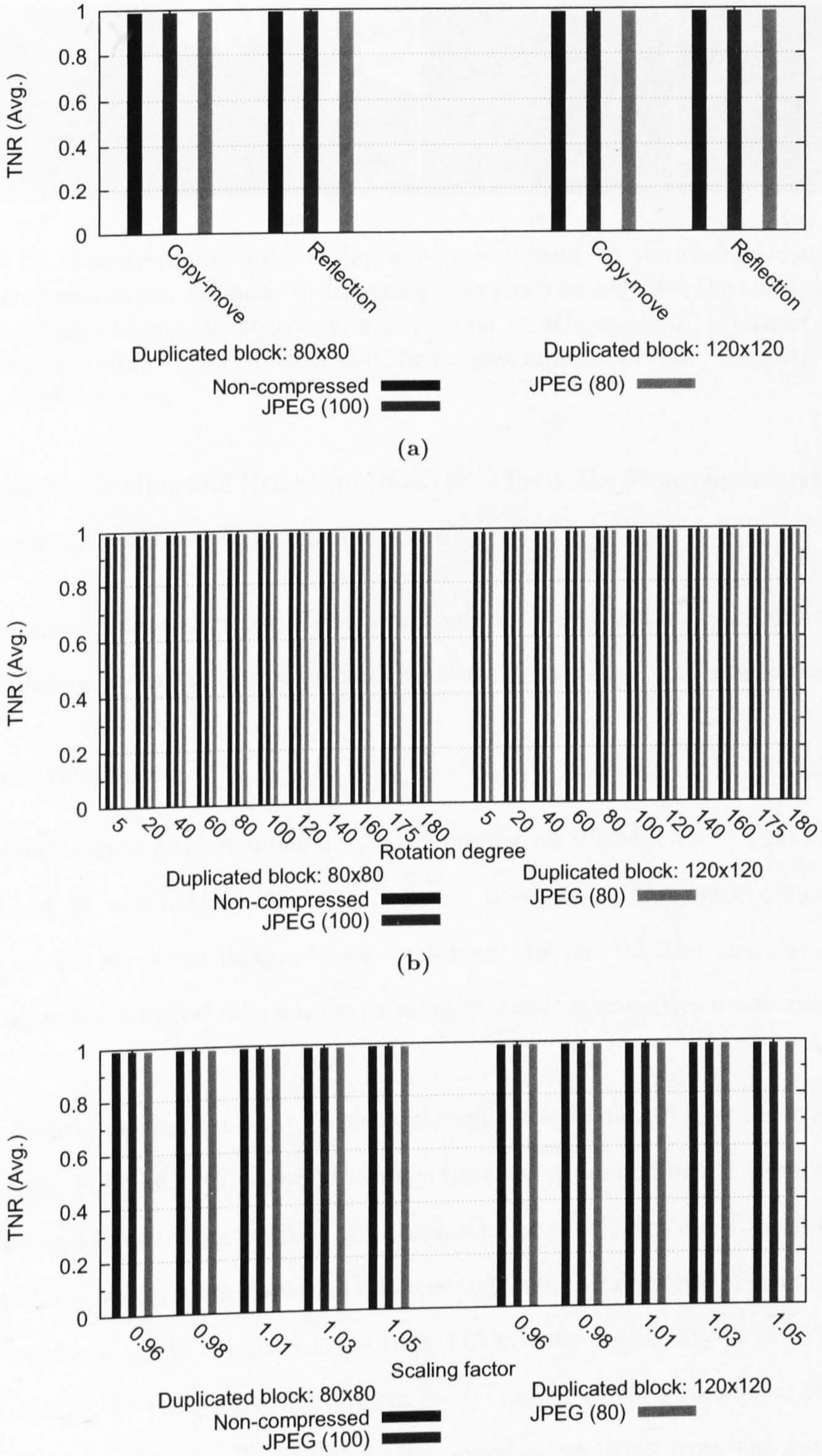


(a)

(b)

(c)

Figure 6.10: TPR results of the extensive tests using a block-size of 32 × 32.



(a)

(b)

(c)

Figure 6.11: TNR results of the extensive tests using a block-size of 32 × 32.

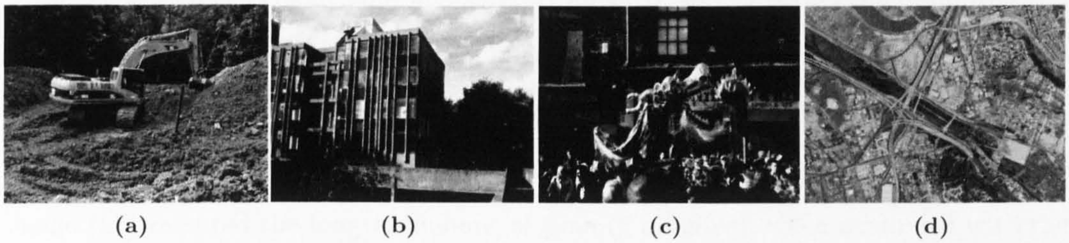


Figure 6.12: The response time of the detectors depend on the characteristics of the image being examined. (a) and (b) Images whose analysis required the shortest and the longest amount of time, respectively, using Myna *et al.*'s method. (c) and (d) Images whose analysis required the shortest and the longest amount of time, respectively, using the proposed detector.

- Rotation, Scaling and Reflection (Rot.+Sc.+Ref.); the same parameters as in the preceding point in addition to reflection.

Thus, seven different forgeries were produced from every test image. All the forgeries –140 in total– were submitted to the two duplicate detectors in non-compressed format.

Computational cost

Both methods were implemented in C++, running on a 32-bit CPU 2.80 GHz, with 1 GB RAM. It was noticed that the response time of the two detectors depend on the characteristics of the images being examined. In general, the analysis of images containing scarce textural information or many intrinsic symmetries tends take a longer time.

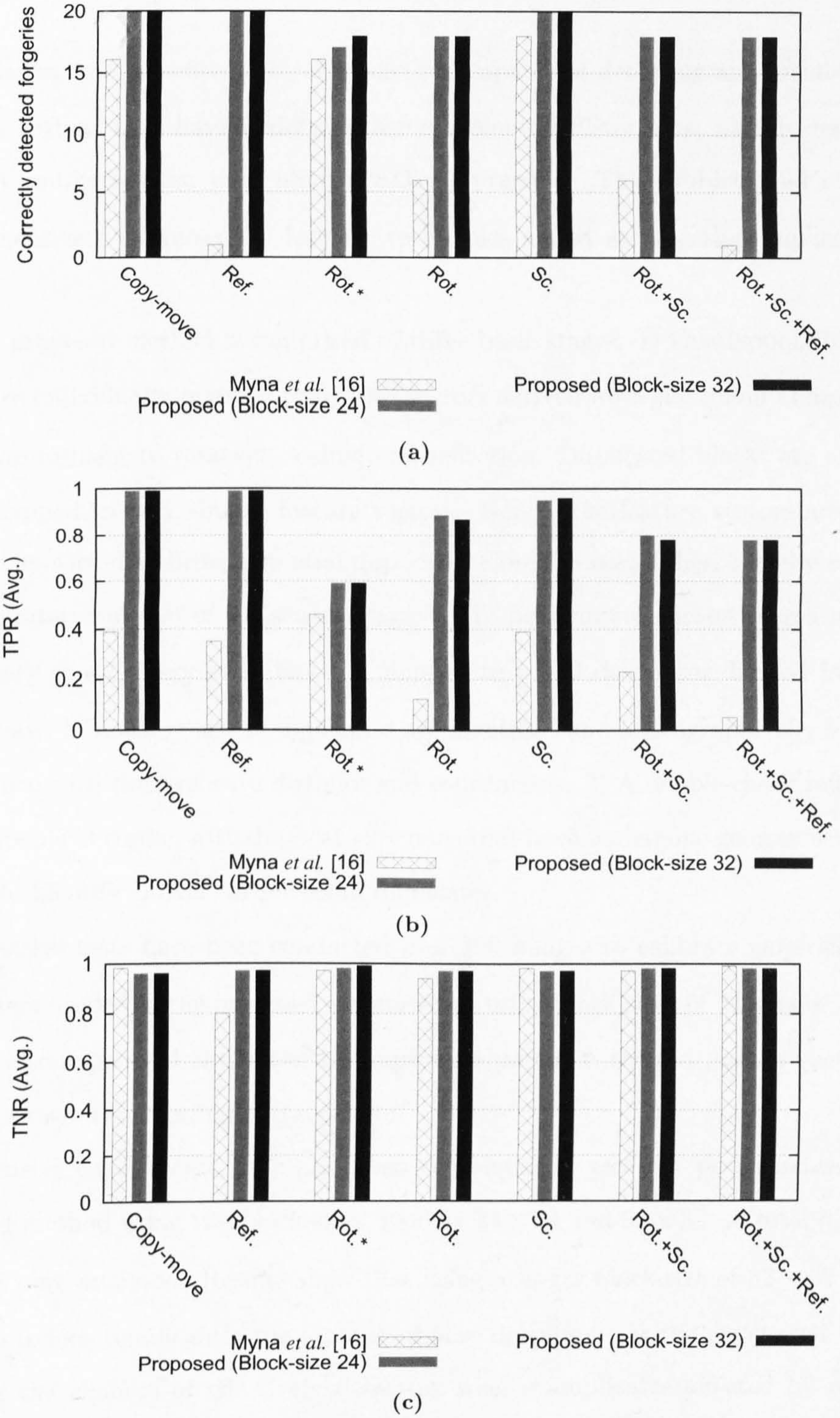
In half of the cases, Myna *et al.*'s method required less than 15 minutes to examine a single image. Figure 6.12(a) shows the image that was examined in the shortest amount of time using Myna *et al.*'s method (87 seconds). However, in some 10% of the cases, Myna *et al.*'s system required over 90 minutes to produce a response. For example, the analysis of the image in Figure 6.12(b) took 115 minutes, using Myna *et al.*'s system. This is because the refinement mechanism in the higher wavelet resolution level failed to discard an important proportion of false matches resulting from the symmetrical structures of the building. As a result, the computational cost of the refinement in the subsequent wavelets coefficients was severely affected. On the other hand, on average, the proposed method examined a single image in less than 2 minutes (using either 24×24 or 32 × 32 blocks). The image whose analysis required the shortest amount of time (45 seconds), using the proposed method, is shown in Figure 6.12(c). Conversely, the image that required the longest amount of time (6 minutes), when examined with the proposed scheme, is shown in Figure 6.12(d). A plausible explanation for the later case is that the textured areas are not sharp enough to survive the loss due to interpolation during the computation of the 1-D descriptors, thereby increasing the number of false matches and the time needed by the refinement stage.

Detection and localisation performance

Myna *et al.*'s method mistakenly regarded 6 original images as containing duplicates (TNR $\neq$ 1). The proposed system, on the other hand, produced 3 false detections, when using 24 × 24 blocks, and only 2, when using 32 × 32 blocks.

Figure 6.13(a) shows the number of forgeries correctly detected (TPR $\neq$ 0). As expected, nearly all the duplicates affected by reflection went undetected by Myna *et al.*'s scheme. Surprisingly, though, this scheme also exhibited serious difficulties in detecting duplicates that have undergone rotations by angles other than 90°, 180° and 270°. In contrast, the proposed method showed a consistent performance, detecting over 85% of the total number of forgeries, regardless the employed block-size.

The average TPRs, calculated over each forgery correctly detected, are presented in Figure 6.13(b). Throughout the tests, the proposed method compares favourably with Myna's method, which encountered serious difficulties in correctly identifying replicated regions rotated by angles non-multiples of 90°. Figure 6.13(c) shows the average TPRs assessed from every forgery correctly detected. The results achieved by the two methods are rather comparable in all the cases, except for the reflection test where Myna *et al.*'s method achieved the lower TNR (0.80).



(a)

(b)

(c)

Figure 6.13: Comparison results. (a) Correctly detected forgeries. (b) Average TPR results. (c) Average TNR results.

6.5 Summary and Conclusions

This chapter has presented a forensic method capable of detecting and localising duplicated regions that have undergone some geometric distortions, namely reflection, rotation and/or scaling, even after JPEG compression. This problem had not been comprehensively addressed by forensic techniques aimed at detecting duplicated regions.

The proposed method is comprised of three basic stages. 1) Overlapping blocks of pixels are individually mapped to feature vectors derived from the colour components, which are resilient to rotation, scaling and reflection. Duplicated blocks are expected to be mapped to very similar feature vectors. Hence, the feature vectors are lexicographically sorted to bring potential duplicates closer to each other, thereby reducing the computational cost of the search stage. 2) To perform an efficient search in terms of memory space, every pixel block is mapped to a 1-D descriptor derived from log-polar maps. Potential pairs of duplicates are identified and lexicographically sorted in accordance with their relative distance and coordinates. 3) A double-check refinement stage capable of coping with duplicated regions that have undergone geometric changes is used to identify clusters of potential duplicates.

Extensive tests have been conducted over 700 images to calibrate empirically the parameters needed in the proposed system when using block-sizes of 24×24 or 32×32. Results also supported the initial assumptions that the extracted feature vectors are resilient to rotation, scaling and reflection.

Extensive experiments have also been conducted to test the performance of the proposed method using two block-sizes, namely 24×24 and 32×32. A total of 10,800 forgeries were analysed. Results show that using a larger block-size of 32×32 is beneficial to reduce significantly the number of false detections, at the expense of slightly reducing the chances of effectively detecting smaller duplicates affected by rotation and/or scaling. Moreover, even though, the difference between the localisation performance achieved when using 24 × 24 is comparable to that achieved when using 32 blocks, using a block-size of 32 × 32 can mitigate the impact that JPEG compression has in the detection performance.

Finally, results showed that the proposed method compares favourably to the existing scheme, proposed by Myna *et al.*, in terms of detection/localisation performance and computational cost.

Chapter 7

Conclusions and Outlook

7.1 Conclusions

As powerful processing technology becomes available in conventional image editing software, the creation of convincing image forgeries has become increasingly easier, even for inexperienced users. This has raised serious concerns about the credibility of digital images, especially in areas that rely on visual information. This thesis investigated this problem and proposed possible solutions based on fragile watermarking and image forensics.

Chapter 4 presents a closer look at the singular properties of an existing fragile watermarking method, which enables the use of the same key to watermark several images, each one associated to a unique image index. An investigation has been presented into the security limitations of such an existing method, in applications that require higher tampering localisation. Then, a method has been proposed to achieve higher tampering localisation without compromising the security of the algorithm. This is achieved by taking a different approach to verification, wherein the bit strings retrieved from every block are examined to identify possible similarities, which are then used to identify authentic pixel blocks. The proposed mechanism renders brute force attacks computationally infeasible when the key space, which is independent of the employed block-size, is chosen to be sufficiently large. Comparison results demonstrated that the localisation capabilities of the proposed method outperforms state-of-the-art schemes when tested against different types of manipulations.

Chapter 5 presents a fragile watermarking method that combines a pixel-wise and a block-wise mechanisms to provide enhanced tampering localisation and exact self-recovery capabilities. The parameters of the proposed method can be adjusted to control the embedding distortion, as well as the probabilities of false validations and false recoveries. Extensive tests, conducted on 900 images, have been presented to evaluate the localisation and restoration performance of the proposed method, in comparison with a state-of-the-art scheme, against tampered regions of a range of sizes with different proportions of altered pixels. Results showed that, when using 2 LSBPs of the image to allocate the watermark, the proposed method achieved a performance comparable to the existing method, which always uses 3 LSBPs, thereby introducing a higher embedding distortion. Even better is the fact that, when using 3 LSBPs, the proposed method clearly outperformed the existing scheme, especially for tampered regions that extended to less than 20% of the image. Additionally, results showed that the proposed method is capable of partially restoring missing regions of watermarked pixels that had been removed by cropping. Resilience to cropping is a problem that had not been addressed by methods with self-recovery capabilities.

Chapter 6 presents a passive forensic method capable of detecting duplicated regions affected by reflection, rotation and/or scaling, even after mild JPEG compression. This problem had not been comprehensively addressed in existing literature. In the proposed algorithm, overlapping pixel blocks are individually mapped to colour-based feature vectors resilient to reflection, rotation and scaling. Thus, the search space for potential duplicates is limited to blocks associated to similar feature vectors. Moreover, to perform an even more efficient search, every pixel block is mapped to a 1-D descriptor derived from log-polar maps. Finally, a refinement mechanism is used to identify clusters of potential duplicates, separated by similar offsets. Extensive tests have been conducted to test the performance of the proposed method using two block-sizes, namely 24 × 24 and 32 × 32. Results showed that the number of false detections was significantly less when using 32 × 32 blocks, at the expense of reducing the chances of detecting smaller duplicates affected by rotation and/or scaling. Furthermore, results showed that, even though that the performance of the method is comparable when using the two tested block-sizes, the effects of JPEG compression are less significant when using 32 × 32 blocks. Results also demonstrated that the proposed method compares favourably to an existing scheme, terms of detection/localisation of duplicates, as well as computational cost.

7.2 Outlook

The investigations presented in this thesis suggest the following directions for future research:

- The self-recovery performance of the fragile watermarking method presented in Chapter 5 could be improved by incorporating a conventional image restoration technique, e.g. inpainting. This way, the current exhaustive search for potential valid pixel values could be limited to a range of expected values derived from genuine pixels in a close neighbourhood. The foreseen impact of this approach is twofold. On the one hand, a larger number of pixels would be effectively restored, while on the other hand, the computational cost of the restoration algorithm would be significantly reduced. This approach is expected to improve the recovery performance, even in cases of images containing scattered distortions, such as salt-and-pepper noise.

- The method presented in Chapter 5 can be used to provide approximate restoration capabilities by recovering wavelet coefficients, instead of pixel values directly. Since every wavelet coefficient is associated to a subset of adjacent pixels in the spatial domain, the suggested approach is expected to increase significantly the percentage of content that can be significantly restored, at the expense of slightly degrading the quality of the recovered image.

- The incorporation of mechanisms resilient to non-obtrusive distortions, such as JPEG compression, to the scheme presented in Chapter 5 would boost the number of applications of the recovery algorithm. To this end, the quantisation-based technique proposed in [140] could be considered a starting point.
- Examining further features in the final refinement stage of the detector of duplicates, presented in Chapter 6, could contribute to a reduction of false alarms. For example, clusters could be identified by considering not only the offset between pairs of potential duplicates, but also the transformation parameters (e.g. scaling factor and rotation angle) that relate them. This could be efficiently achieved by incorporating a set of Speed Up Robust Features (SURF), described in [141].

Bibliography

[1] G. Friedman, "The trustworthy digital camera: restoring credibility to the photographic image," *IEEE Transactions on Consumer Electronics*, vol. 39, no. 4, pp. 905 – 910, 1993.

[2] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker, *Digital watermarking and steganography*, 2nd ed. Morgan Kauffman, 2008.

[3] M. Holliman and N. Memon, "Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes," *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 432 – 441, 2000.

[4] P. S. L. M. Barreto, H. Y. Kim, and V. Rijmen, "Toward secure public-key blockwise fragile authentication watermarking," in *Prof. ICIP – IEEE of International Conference on Image Processing*, vol. 2, 2002, pp. 57 – 62.

[5] J. Fridrich, D. Soukal, and J. Lukáš, "Detection of copy-move forgery in digital images," in *Proc. of Digital Forensic Research Workshop*, August 2003, pp. 55–61.

[6] S. Katzenbeisser and A. P. F. Peticolas, Eds., *Information Hiding Techniques for Steganography and Digital Watermarking.* Artech Print on Demand, Dec 1999.

[7] C. Rey and J.-L. Dugelay, "A survey of watermarking algorithms for image authentication," *Eurasip Journal on Applied Signal Processing*, vol. 2002, no. 6, pp. 613 – 621, 2002.

[8] G. Coatrieux, H. Maitre, B. Sankur, Y. Rolland, and R. Collorec, "Relevance of watermarking in medical imaging," in *Proc. of ITAB – IEEE/EMBS International Conference on Information Technology Applications in Biomedicine*, Arlington, VA, 2000, pp. 250 – 255.

[9] Y. Q. Shi, "Reversible data hiding," in *Proc. of IWIH – International Workshop on Information Hiding*, vol. 3304, Seoul, South Korea, 2005, pp. 1 – 12.

[10] J. Lee and C. S. Won, "Authentication and correction of digital watermarking images," *Electronics Letters*, vol. 35, no. 11, pp. 886–887, 1999.

[11] ——, "Image integrity and correction using parities of error control coding," in *Proc. of ICME – IEEE International Conference on Multimedia and Expo*, vol. 3, 2000, pp. 1297–1300.

[12] J. Fridrich and M. Goljan, "Protection of digital images using self embedding," in *Proc. of Symposium on Content Security and Data Hiding in Digital Media*, May 1999.

[13] A. H. Paquet, R. K. Ward, and I. Pitas, "Wavelet packets-based digital watermarking for image verification and authentication," *Signal Processing*, vol. 83, no. 10, pp. 2117–2132, 2003.

[14] H. Yang and X. Sun, "Semi-fragile watermarking for image authentication and tamper detection using HVS model," in *Proc. of International Conference on Multimedia and Ubiquitous Engineering*, vol. 0. IEEE Computer Society, 2007, pp. 1112–1117.

[15] M. Kutter and F. Hartung, *Introduction to Watermarking Techniques*, 1st ed., ser. Computer Security Series. Artech House, 2000, ch. 5, pp. 97–120.

[16] H. Knowles, D. Winne, C. Canagarajah, and D. Bull, "Image tamper detection

and classification using support vector machines," *IEE Proc. – Vision, Image and Signal Processing*, vol. 151, no. 4, pp. 322–328, 2004.

[17] X. Li, D. Tao, X. Gao, and W. Lu, "A natural image quality evaluation metric," *Signal Processing*, vol. 89, no. 4, pp. 548–555, 2009.

[18] W. Lu, X. Li, X. Gao, W. Tang, J. Li, and D. Tao, "A video quality assessment metric based on human visual system," *Cognitive Computation*, vol. 2, no. 2, pp. 120–131, 2010.

[19] M. Sampat, Z. Wang, S. Gupta, A. Bovik, and M. Markey, "Complex wavelet structural similarity: A new image similarity index," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2385–2401, 2009.

[20] J. Fridrich, "Security of fragile authentication watermarks with localization," in *Proc. of Security and Watermarking of Multimedia Contents*, vol. 4675. CA, USA: SPIE, 2002, pp. 691 – 700.

[21] B. B. Zhu, M. D. Swanson, and A. H. Tewfik, "When seeing isn't believing," *IEEE Signal Processing Magazine*, vol. 21, no. 2, pp. 40 – 49, 2004.

[22] I. Venturini, "Counteracting oracle attacks," in *Proc. of MM&SEC – Workshop on Multimedia and Security.* Magdeburg, Germany: ACM, 2004, pp. 187 – 192.

[23] P. W. Wong, "A public key watermark for image verification and authentication," in *Proc. of ICIP – IEEE International Conference on Image Processing*, vol. 1, Chicago, IL, USA, 1998, pp. 455 – 459.

[24] P. Wong and N. Memon, "Secret and public key image watermarking schemes for image authentication and ownership verification," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1593 – 1601, 2001.

[25] A. H. Ouda and M. R. El-Sakka, "A practical version of wong's watermarking

technique," in *Proc. of ICIP- IEEE Internation Conference on Image Processing*, 2004, pp. 2615–2618.

[26] S. Suthaharan, "Fragile image watermarking using a gradient image for improved localization and security," *Pattern Recognition Letters*, vol. 25, no. 16, pp. 1893 – 1903, 2004.

[27] M. Celik, G. Sharma, E. Saber, and A. Takalp, "A hierarchical image authentication watermark with improved localization and security," in *Proc. of ICIP – IEEE International Conference on Image Processing*, vol. 2, Thessaloniki, 2001, pp. 502 – 505.

[28] M. U. Celik, G. Sharma, E. Saber, and A. M. Tekalp, "Hierarchical watermarking for secure image authentication with localization," *IEEE Transactions on Image Processing*, vol. 11, no. 6, pp. 585 – 595, 2002.

[29] C.-T. Li and H. Si, "Wavelet-based fragile watermarking scheme for image authentication," *Journal of Electronic Imaging*, vol. 16, no. 1, pp. 1–9, 2007.

[30] H. He, J. Zhang, and H.-M. Tai, "A wavelet-based fragile watermarking scheme for secure image authentication," in *Proc. of IWDW – International Workshop on Digital Watermarking*, vol. 4283, 2006, pp. 422 – 432.

[31] X. Zhang and S. Wang, "Statistical fragile watermarking capable of locating individual tampered pixels," *IEEE Signal Processing Letters*, vol. 14, no. 10, pp. 727–730, 2007.

[32] Y.-L. You and M. Kaveh, "Fourth-order partial differential equations for noise removal," *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 1723–1730, 2000.

[33] C. Brito-Loeza and K. Chen, "On high-order denoising models and fast algorithms

for vector-valued images," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1518–1527, 2010.

[34] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 882–889, 2003.

[35] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1200–1212, 2004.

[36] M. Yeung and F. Mintzer, "An invisible watermarking technique for image verification," in *Proc. of ICIP – IEEE International Conference on Image Processing*, vol. vol.2, Santa Barbara, CA, USA, Oct. 1997, pp. 680 – 3.

[37] J. Fridrich, M. Goljan, and N. Memon, "Further attacks on Yeung-Mintzer fragile watermarking scheme," in *Proc. of Security and Watermarking of Multimedia Contents*, vol. 3971. CA, USA: SPIE, 2000, pp. 428 – 437.

[38] J. Fridrich, M. Goljan, and N. Memon, "Cryptanalysis of the Yeung-Mintzer fragile watermarking technique," *Journal of Electronic Imaging*, vol. 11, no. 2, pp. 262–274, 2002.

[39] J. Fridrich, M. Goljan, and A. Baldoza, "New fragile authentication watermark for images," in *Proc. of ICIP – IEEE International Conference on Image Processing*, vol. 1, Vancouver, BC, 2000, pp. 446 – 449.

[40] H. Lu, R. Shen, and F.-L. Chung, "Fragile watermarking scheme for image authentication," *Electronics Letters*, vol. 39, no. 12, pp. 898 – 900, 2003.

[41] S.-H. Liu, H.-X. Yao, W. Gao, and Y.-L. Liu, "An image fragile watermark scheme based on chaotic image pattern and pixel-pairs," *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 869–882, 2007.

[42] H.-J. He, J.-S. Zhang, and H.-M. Tai, “Block-chain based fragile watermarking scheme with superior localization,” in *Proc. of IH – Information Hiding*, 2008, pp. 147–160.

[43] C. W. Hosinger, P. Jones, M. Rabbani, and J. C. Stoffel, “Lossless recovery of an original image containing embedded data,” US Pattent application, Docket No: 77102/E-D, 1999.

[44] J. Fridrich, M. Goljan, and R. Du, “Invertible authentication,” in *Proc. of Security Watermarking of Multimedia Contents*, vol. 4314. CA, USA: SPIE, 2001, pp. 197 – 208.

[45] M. Goljan, J. Fridrich, and R. Du, “Distortion-free data embedding for images,” in *Proc. of IH – Information Hiding*, 2001, pp. 27–41.

[46] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, “Reversible data hiding,” in *Proc of ICIP – IEEE International Conference on Image Processing*, vol. 2, Rochester, NY, United States, 2002, pp. 157–160.

[47] ——, “Lossless generalized-lsb data embedding,” *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 253 – 266, 2005.

[48] M. Celik, G. Sharma, A. Tekalp, and E. Saber, “Localized lossless authentication watermark (LAW),” in *Proc. of Security and Watermarking of Multimedia Contents*, vol. 5020. CA, USA: SPIE, 2003, pp. 689 – 698.

[49] M. U. Celik, G. Sharma, and A. M. Tekalp, “Lossless watermarking for image authentication: A new framework and an implementation,” *IEEE Transactions on Image Processing*, vol. 15, no. 4, pp. 1042 – 1049, 2006.

[50] G. Xuan, J. Zhu, J. Chen, Y. Shi, Z. Ni, and W. Su, “Distortionless data hiding based on integer wavelet transform,” *Electronics Letters*, vol. 38, no. 25, pp. 1646–1648, 2002.

[51] G. Xuan, Y. Shi, Z. Ni, J. Chen, C. Yang, Y. Zhen, and J. Zheng, "High capacity lossless data hiding based on integer wavelet transform," in *Proc. of ISCAS – International Symposium on Circuits and Systems*, vol. 2, 2004, pp. 29–32.

[52] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding-new paradigm in digital watermarking," *Eurasip Journal on Applied Signal Processing*, vol. 2002, no. 2, pp. 185 – 196, 2002.

[53] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890 – 896, 2003.

[54] A. Alattar, "Reversible watermark using difference expansion of quads," in *Proc. of ICASSP – IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, 2004, pp. 377–380.

[55] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147 – 1156, 2004.

[56] J. Stach and A. M. Alattar, "A high-capacity invertible data-hiding algorithm using a generalized reversible integer transform," in *Proc. of Security, Steganography, and Watermarking of Multimedia Contents*. San Jose, CA, USA: SPIE, 2004, pp. 386–396.

[57] L. Kamstra and H. Heijmans, "Reversible data embedding into images using wavelet techniques and sorting," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2082–2090, 2005.

[58] I. F. Kallel, M. S. Bouhlel, and J.-C. Lapayre, "Improved Tian's method for medical image reversible watermarking," *ICGST International Journal on Graphics, Vision and Image Processing*, vol. 7, pp. 1–5, 2007.

[59] Z. Ni, Y. Shi, N. Ansari, and W. Su, "Reversible data hiding," in *Proc. of ISCAS – the International Symposium on Circuits and Systems*, vol. 2, 2003, pp. 912–915.

[60] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354 – 361, 2006.

[61] J. Hwang, J. Kim, and J. Choi, "A reversible watermarking based on histogram shifting," in *Proc. of IWDW – International Workshop on Digital Watermarking*, Jeju Island, Korea, 2006, pp. 348–361.

[62] K.-S. Kim, M.-J. Lee, H.-K. Lee, and Y.-H. Suh, "Histogram-based reversible data hiding technique using subsampling," in *Proc. of MM&SEC – Workshop on Multimedia and Security*. ACM, 2008, pp. 69–74.

[63] F. H. Yeh, G. C. Lee, and C. C. Chiang, "Reversible fragile watermarking based on pyramidal structure and gradient predicting image," in *Proc. of International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, vol. 1. IEEE Computer Society, 2007, pp. 311–314.

[64] R. E. Blahut, *Theory and Practice of Error Control Codes*. Addison-Wesley, 1984.

[65] X. Zhang and S. Wang, "Fragile watermarking scheme using a hierarchical mechanism," *Signal Processing*, vol. 89, no. 4, pp. 675–679, 2009.

[66] X. Zhang, S. Wang, Z. Qian, and G. Feng, "Self-embedding watermark with flexible restoration quality," *Multimedia Tools and Applications.*, 2010, DOI 10.1007/s11042-010-0541-z.

[67] X. Zhang and S. Wang, "Fragile watermarking with error-free restoration capability," *IEEE Transactions on Multimedia*, vol. 10, no. 8, pp. 1490–1499, 2008.

[68] P.-L. Lin, C.-K. Hsieh, and P.-W. Huang, "A hierarchical digital watermarking method for image tamper detection and recovery," *Pattern Recognition*, vol. 38, no. 12, pp. 2519 – 2529, 2005.

[69] C.-C. Chang, Y.-H. Fan, and W.-L. Tai, "Four-scanning attack on hierarchical digital watermarking method for image tamper detection and recovery," *Pattern Recognition*, vol. 41, no. 2, pp. 654–661, 2008.

[70] M.-S. Wang and W.-C. Chen, "A majority-voting based watermarking scheme for color image tamper detection and recovery," *Computer Standards & Interfaces*, vol. 29, no. 5, pp. 561–570, 2007.

[71] H. He and J. Zhang, "Cryptanalysis on majority-voting based self-recovery watermarking scheme," in *Proc. of International Conference on Multimedia Information Networking and Security*, vol. 1. IEEE Computer Society, 2009, pp. 573–577.

[72] T. Y. Lee and S. D. Lin, "Dual watermark for image tamper detection and recovery," *Pattern Recognition*, vol. 41, no. 11, pp. 3497–3506, 2008.

[73] X. Zhang, S. Wang, and G. Feng, "Fragile watermarking scheme with extensive content restoration capability," in *Proc. of IWDW – International Workshop on Digital Watermarking*, 2009, pp. 268–278.

[74] H. Farid, "Seeing is not believing," *IEEE Spectrum*, vol. 46, no. 8, pp. 44–51, 2009.

[75] D. A. Brugioni, *Photo Fakery: The History and Techniques of Photographic Deception and Manipulation*, illustrated ed. Brassey's Inc, 1999.

[76] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," Department of Computer Science, Dartmouth College, Tech. Rep., 2004.

[77] T. Gloe, M. Kirchner, A. Winkler, and R. Böhme, "Can we trust digital image forensics?" in *Proc. of the International Conference on Multimedia.* New York, NY, USA: ACM, 2007, pp. 78–86.

[78] M. Kirchner and R. Böhme, "Tamper hiding: Defeating image forensics," in *Proc. of IH – Information Hiding*, Saint Malo, France, 2007, pp. 326–341.

[79] W. Luo, J. Huang, and G. Qui, "Robust detection of region-duplication forgery in digital images," in *Proc. of ICPR – IAPR International Conference on Pattern Recognition*, vol. 4, 2006, pp. 746–749.

[80] A. Langille and M. Gong, "An efficient match-based duplication detection algorithm," in *Proc. of the 3rd Canadian Conference on Computer and Robot Vision*, 2006, pp. 64–71.

[81] B. Mahdian and S. Saic, "Detection of copy-move forgery using a method based on blur moment invariants," *Forensic Science International*, vol. 171, no. 2-3, pp. 180–189, 2007.

[82] B. Dybala, B. Jennings, and D. Letscher, "Detecting filtered cloning in digital images," in *Proc. of MM&SEC – Workshop on Multimedia and Security.* Dallas, TX, USA: ACM, September 2007, pp. 43–50.

[83] A. N. Myna, M. G. Venkateshmurthy, and C. G. Patil, "Detection of region duplication forgery in digital images using wavelets and log-polar mapping," in *Proc. of the International Conference on Computational Intelligence and Multimedia Applications*, Washington, DC, USA, 2007, pp. 371–377.

[84] H.-J. Lin, C.-W. Wang, and Y.-T. Kao, "Fast copy-move forgery detection," *WSEAS Trans. Signal Processing*, vol. 5, no. 5, pp. 188–197, 2009.

[85] J. Wang, G. Liu, H. Li, Y. Dai, and Z. Wang, "Detection of image region duplication forgery using model with circle block," in *Proc. of International Conference*

on Multimedia Information Networking and Security, vol. 1. IEEE Computer Society, 2009, pp. 25–29.

[86] S. Bayram, H. Sencar, and N. Memon, “An efficient and robust method for detecting copy-move forgery,” in *Proc. of ICASSP – IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 1053–1056.

[87] H. Huang, W. Guo, and Y. Zhang, “Detection of copy-move forgery in digital images using SIFT algorithm,” in *Proc. of IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, vol. 2, 2008, pp. 272–276.

[88] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, “Geometric tampering estimation by means of a SIFT-based forensic analysis,” in *Proc. of ICASSP – IEEE International Conference on Acoustics Speech and Signal Processing*, 2010, pp. 1702–1705.

[89] X. Pan and S. Lyu, “Detecing image region duplication using SIFT features,” in *Proc. of ICASSP – IEEE International Conference on Acoustics, Speech, and Signal Processing*, Dallas, TX, USA, 2010, pp. 1706–1709.

[90] S.-J. Ryu, M.-J. Lee, and H.-K. Lee, “Detection of copy-rotate-move forgery using zernike moments,” in *Proc. of IH – Information Hiding*, 2010, pp. 51–65.

[91] A. C. Popescu and H. Farid, “Exposing digital forgeries by detecting traces of resampling,” *IEEE Transactions on Signal Processing*, vol. 53, no. 2-2, pp. 758–767, 2005.

[92] M. Kirchner, “Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue,” in *Proc. of MM&SEC – Workshop on Multimedia and Security*. Oxford, United Kingdom: ACM, 2008, pp. 11–20.

[93] A. Gallagher, “Detection of linear and cubic interpolation in jpeg compressed

images," in *Proc. of the Canadian Conference on Computer and Robot Vision*, 2005, pp. 65–72.

[94] B. Mahdian and S. Saic, "Blind authentication using periodic properties of interpolation," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 529–538, 2008.

[95] M. Kirchner and R. Bohme, "Hiding traces of resampling in digital images," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 4, pp. 582–592, 2008.

[96] T.-T. Ng and S.-F. Chang, "A model for image splicing," in *Proc. of ICIP – IEEE International Conference on Image Processing*, vol. 2, 2004, pp. 1169–1172.

[97] H. Farid, "Detecting digital forgeries using bispectral analysis," AI Lab, Masachusetts Institute of Technology, Tech. Rep. AIM-1657, 1999.

[98] T.-T. Ng, S.-F. Chang, and Q. Sun, "Blind detection of photomontage using higher order statistics," in *Proc. of ISCAS – International Symposium on Circuits and Systems*, vol. 5, 2004, pp. 688–691.

[99] Y. Q. Shi, C. Chen, and W. Chen, "A natural image model approach to splicing detection," in *Proc. of MM&SEC – Workshop on Multimedia and Security*. ACM, 2007, pp. 51–62.

[100] Z. Qu, G. Qiu, and J. Huang, "Detect digital image splicing with visual cues," in *Proc. of IH – Information Hiding*, vol. 5806, 2009, pp. 247–261.

[101] R. C. Gonzalez and R. E. Woods, *Digital image processing.* Prentice Hall, 2008.

[102] J. Lukáš and J. Fridrich, "Estimation of primary quantization matrix in double compressed JPEG images," *Proc. of DFRWS – Digital Forensic Research Workshop*, 2003.

[103] A. C. Popescu and H. Farid, "Statistical tools for digital forensics," in *Proc. of IWIH – International Workshop on Information Hiding*, vol. 3200, 2004, pp. 128–147.

[104] D. Fu, Y. Q. Shi, and W. Su, "A generalized benford's law for JPEG coefficients and its applications in image forensics," in *Proc. of Security, Steganography, and Watermarking of Multimedia Contents*, vol. 6505. SPIE, Feb 2007, pp. 1–11.

[105] B. Li, Y. Shi, and J. Huang, "Detecting doubly compressed JPEG images by using mode based first digit features," in *Proc. of IEEE Workshop on Multimedia Signal Processing*, 2008, pp. 730–735.

[106] B. Mahdian and S. Saic, "Detecting double compressed JPEG images," in *Proc. of ICDP – International Conference on Crime Detection and Prevention*, 2009, pp. 1–6.

[107] X. Feng and G. Doerr, "JPEG recompression detection," in *Proc. of Media Forensics and Security*, vol. 7541. SPIE, Feb 2010, pp. 1–12.

[108] Y. Shi, "First digit law and its application to digital forensics," in *Proc. of IWDW – International Workshop on Digital Watermarking*, Busan, Korea, 2009, pp. 448–453.

[109] J. Wang, B.-H. Cha, S.-H. Cho, and C.-C. Kuo, "Understanding benford's law and its vulnerability in image forensics," in *Proc. of ICME – IEEE International Conference on Multimedia and Expo*, 2009, pp. 1568–1571.

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

[111] H. Farid, "Digital image ballistics from JPEG quantization," Department of Computer Science, Dartmouth College, Tech. Rep. TR2008-638, 2008.

[112] J. D. Kornblum, "Using JPEG quantization tables to identify imagery processed by software," *Digital Investigation*, vol. 5, no. Supplement 1, pp. 21–25, Sep 2008.

[113] J. He, Z. Lin, L. Wang, and X. Tang, "Detecting doctored JPEG images via DCT coefficient analysis," in *Proc. of ECCV – European Conference on Computer Vision*, vol. 3953. Graz. Austria: Springer Berlin / Heidelberg, 2006, pp. 423–435.

[114] Z. Lin, J. He, X. Tang, and C.-K. Tang, "Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis," *Pattern Recognition*, vol. 42, no. 11, pp. 2492–2501, 2009.

[115] W. Luo, Z. Qu, J. Huang, and G. Qiu, "A novel method for detecting cropped and recompressed image block," in *Proc. of ICASSP – IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, 2007, pp. 217–220.

[116] S. Ye, Q. Sun, and E.-C. Chang, "Detecting digital image forgeries by measuring inconsistencies of blocking artifact," in *Proc. of ICME – IEEE International Conference on Multimedia and Expo*, 2007, pp. 12–15.

[117] M. Kirchner and T. Gloe, "On resampling detection in re-compressed images," in *Proc. of WIFS – IEEE International Workshop on Information Forensics and Security*, 2009, pp. 21–25.

[118] Z. Lin, R. Wang, X. Tang, and H.-Y. Shum, "Detecting doctored images using camera response normality and consistency," in *Proc. of CVPR – IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 1087–1092.

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

[120] ——, "Image splicing detection using camera response function consistency and automatic segmentation," in *Proc. of ICME – IEEE International Conference on Multimedia and Expo*, 2007, pp. 28–31.

[121] A. Gallagher and T. Chen, "Image authentication by detecting traces of demosaicing," in *Proc. of CVPRW – IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2008, pp. 1–8.

[122] Y. Huang, "Can digital image forgery detection be unevadable? a case study: color filter array interpolation statistical feature recovery," in *Proc. of Visual Communications and Image Processing*, vol. 5960. SPIE, Jul 2005, pp. 1–12.

[123] M. Kirchner and R. Bohme, "Synthesis of color filter array pattern in digital images," in *Proc. of Media Forensics and Security*, vol. 7254. SPIE, Feb 2009, pp. 1–14.

[124] M. K. Johnson and H. Farid, "Exposing digital forgeries through chromatic aberration," in *Proc. of MM&SEC – Workshop on Multimedia and Security.* ACM, 2006, pp. 48–55.

[125] T. Gloe, K. Borowka, and A. Winkler, "Efficient estimation and large-scale evaluation of lateral chromatic aberration for digital image forensics," in *Proc. of Media Forensics and Security*, vol. 7541. SPIE, Feb 2010, pp. 1–13.

[126] M. K. Johnson and H. Farid, "Exposing digital forgeries by detecting inconsistencies in lighting," in *Proc. of MM&SEC – Workshop on Multimedia and Security.* ACM, 2005, pp. 1–10.

[127] P. Nillius and J. Eklundh, "Automatic estimation of the projected light source direction," in *Proc. of CVPR – IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. 1076–1083.

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

[129] ——, "Exposing digital forgeries through specular highlights on the eye," in *Proc. of IH – Information Hiding*, Saint Malo, France, 2007, pp. 311–325.

[130] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd ed. New York, NY, USA: John Wiley & Sons, Inc., 1996.

[131] W. Zeng, B. Liu, and S. Lei, "Extraction of multiresolution watermark images for resolving rightful ownership," in *Proc. of Security and Watermarking of Multimedia Contents*, vol. 3657. SPIE, Apr 1999, pp. 404–414.

[132] R. Rivest, "The MD5 Message-Digest Algorithm," United States, 1992.

[133] *Secure Hash Standard.* Washington: National Institute of Standards and Technology, 2002, federal Information Processing Standard 180-2. [Online]. Available: http://csrc.nist.gov/publications/fips/

[134] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot, *Handbook of Applied Cryptography.* CRC Press, Inc., 1996.

[135] M. H. Degroot and M. J. Schervish, *Probability and Statistics*, 1st ed. Addison Wesley, 1975.

[136] N. Memon and P. W. Wong, "A buyer-seller watermarking protocol," *IEEE Transactions on Image Processing*, vol. 10, no. 4, pp. 643–649, 2001.

[137] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Institute of Technology, Tech. Rep. 7694, 2007. [Online]. Available: http://authors.library.caltech.edu/7694

[138] C.-Y. Lin, M. Wu, J. Bloom, I. Cox, M. Miller, and Y. Lui, "Rotation, scale, and translation resilient watermaking for images," *IEEE Transactions on Image Processing*, vol. 10, no. 5, pp. 767 – 782, 2001.

[139] R. N. Bracewell, *The Fourier transform and its application.* McGraw-Hill, 1986.

[140] C. Lin and S. Chang, "Semi-fragile watermarking for authenticating JPEG visual content," in *Proc. of SPIE Security and Watermarking of Multimedia Content*, San Jose, CA, USA, January 2000, pp. 140–151.

[141] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.