



University of Bradford eThesis

This thesis is hosted in [Bradford Scholars](#) – The University of Bradford Open Access repository. Visit the repository for full metadata or to contact the repository team



© University of Bradford. This work is licenced for reuse under a [Creative Commons Licence](#).

**DESIGN AND ANALYSIS OF DISCRETE COSINE
TRANSFORM-BASED WATERMARKING
ALGORITHMS FOR DIGITAL IMAGES**

Development and evaluation of blind Discrete Cosine Transform-Based watermarking algorithms for copyright protection of digital images using handwritten signatures and mobile phone numbers

AHMED M. N. AL-GINDY

Submitted for the degree of Doctor of Philosophy

School of Computing, Informatics and Media

University of Bradford

2011

ABSTRACT

DESIGN AND ANALYSIS OF DISCRETE COSINE TRANSFORM-BASED WATERMARKING ALGORITHMS FOR DIGITAL IMAGES

Ahmed M.N. Al-Gindy

Keywords

Image Processing, watermarking, Discrete Cosine Transform (DCT), Still Grey-scale
Images, Still Colour Images.

This thesis deals with the development and evaluation of blind discrete cosine transform-based watermarking algorithms for copyright protection of digital still images using handwritten signatures and mobile phone numbers. The new algorithms take into account the perceptual capacity of each low frequency coefficients inside the Discrete Cosine Transform (DCT) blocks before embedding the watermark information. They are suitable for grey-scale and colour images. Handwritten signatures are used instead of pseudo random numbers. The watermark is inserted in the green channel of the RGB colour images and the luminance channel of the YCrCb images. Mobile phone numbers are used as watermarks for images captured by mobile phone cameras. The information is embedded multiple-times and a shuffling scheme is applied to ensure that no spatial correlation exists between the original host image and the multiple watermark copies. Multiple embedding will increase the robustness of the watermark against attacks since each watermark will be individually reconstructed and verified before applying an averaging process. The averaging process has managed to reduce the amount of errors of the extracted information. The developed watermarking methods are shown to be robust against JPEG compression, removal attack, additive noise, cropping, scaling, small degrees of rotation, affine, contrast enhancements, low-pass, median filtering and Stirmark

attacks. The algorithms have been examined using a library of approximately 40 colour images of size 512×512 with 24 bits per pixel and their grey-scale versions. Several evaluation techniques were used in the experiment with different watermarking strengths and different signature sizes. These include the peak signal to noise ratio, normalized correlation and structural similarity index measurements. The performance of the proposed algorithms has been compared to other algorithms and better invisibility qualities with stronger robustness have been achieved.

Acknowledgments

I would like to thank the following persons who supported me during the time that I was working for this Degree.

I wish to express my deepest gratitude to my supervisors, Dr Rami Qahwaji and Prof. Hussain Al-Ahmad, for their continues advice and support. My supervisor's responsibilities did not prevent them from offering me advice and support, whenever I needed them. They did the best to provide me with all facilities that I needed. The continuous supervision and friendly discussions that I had with my supervisors had great influence on me and on this work.

A very important person to thank is Dr. Ayman Tawfik. The friendly and encouraging attitude of Dr. Ayman is gratefully acknowledged. I visited him many times and stayed in his office for long periods, and every time he did his best to provide me with all help that I needed.

My greatest acknowledgements are to my parents and family members for being very close with their support .

Contents

Contents	5
List of Figures.....	9
List of Tables	10
List of Abbreviations	13
List of Variables	14
Chapter 1 Introduction.....	16
1.1 The Importance of Digital Image Watermarking	16
1.2 Aims and Contributions	17
1.3 Applications of Watermarking	18
1.3.1 Copyright Owner Identification	18
1.3.2 Fingerprinting or Transaction Tracking	19
1.3.3 Broadcast Monitoring.....	19
1.3.4 Data Hiding	19
1.3.5 Data Authentication.....	19
1.3.6 Copy Control	20
1.3.7 Device Control	20
1.4 Watermarking Classifications.....	20
1.4.1 Invisible versus Visible Watermarking	20
1.4.2 Blind versus Non-Blind Watermarking	21
1.4.3 Robust versus Fragile Watermarking	21
1.5 Watermarking Requirements	21
1.5.1 Perceptual Transparency or Imperceptibility.....	21
1.5.2 Payload	22
1.5.3 Robustness.....	22
1.5.4 Security	22
1.5.5 Trustworthiness	22

1.6	Some Significant Known Attacks.....	23
1.6.1	JPEG Compression Attack	23
1.6.2	Image Enhancement Operations.....	24
1.6.3	Removal Attack.....	25
1.6.4	Cropping Attack	25
1.6.5	Additive Noise.....	26
1.6.6	Resize Attack.....	26
1.6.7	Filtering	27
1.6.8	Standard Assessment Tools.....	27
1.7	Challenges in Digital Watermarking	29
1.8	Overview of the Thesis.....	30
Chapter 2	Digital Image Watermarking.....	31
2.1	Overview	31
2.2	Evaluation and Benchmarking of Watermarking Systems	31
2.2.1	Watermarking Datasets	32
2.2.2	Performance Evaluation and Representation.....	32
2.3	Literature Survey of Watermarking Techniques	34
2.3.1	Spatial Domain Techniques.....	35
2.3.2	Frequency Domain Techniques	37
2.3.3	Watermarking Techniques using mobile devices	42
2.4	Final Remarks	43
Chapter 3	Digital Watermarking Algorithms for Grey-scale Images.....	46
3.1	Overview	46
3.2	Host Images and Watermarks	46
3.3	Performance Evaluation and Testing Strategy.....	48
3.4	Algorithm 1: A Blind Image Watermarking of Handwritten Signatures Using Low-Frequency Band DCT Coefficients	49
3.4.1	Proposed Robust Image Watermark Algorithm	50
3.4.2	The Embedding Process	50
3.4.3	The shuffle Scheme	53

3.4.4	The Reconstruction Process	55
3.4.5	Results	57
3.5	Algorithm 2: An Adaptive Secured Watermarking Algorithm	66
3.5.1	The DCT Coefficients Selection (DCS) Process	67
3.5.2	Results	70
3.6	Algorithm 3: Blind Image Watermarking for High capacity watermarks Using Low-Frequency Band DCT Coefficients	79
3.6.1	The Proposed High Capacity Image Watermarking Algorithm	79
3.6.2	Results	81
3.7	Comparison with previous work.....	91
3.8	Final Remarks	93
Chapter 4	Digital Watermarking Algorithms for Colour Images.....	94
4.1	Overview	94
4.2	Algorithm 4: Watermarking of Colour Images in the DCT Domain Using the Y Channel	94
4.2.1	Embedding and Extraction Algorithms	94
4.2.2	Simulation and Results	96
4.3	Algorithm 5: A Novel Blind Image Watermarking Technique for Colour RGB Images in the DCT Domain Using Green Channel.....	105
4.3.1	Analysis of Colour Images	105
4.3.2	The Embedding and Extraction Algorithms	107
4.3.3	Simulation and Results	110
4.4	Algorithm 6: A High Capacity Watermarking Technique for the Copyright protection of Colour Images	119
4.4.1	Embedding and Extraction Steps.....	119
4.4.2	Simulation and Results	121
4.5	Comparison with previous work.....	132
4.6	Final Remarks	134
Chapter 5	Watermarking Algorithms for Images Captured by Mobile Phone Cameras	136
5.1	Overview	136

5.2	Algorithm 7: Blind Image Watermarking of Mobile Phone Numbers Using Low-Frequency DCT Coefficients	136
5.2.1	The Proposed Algorithm	137
5.2.2	The DCT Selection Coefficients (DCS) Process	138
5.2.3	Embedding and Extraction Steps.....	140
5.2.4	Results	142
5.3	Comparison with Previous Work	149
5.4	Final Remarks	149
Chapter 6	Conclusions and Recommendations	150
6.1	Overview	150
6.2	Summary of the work	150
6.3	Conclusions	151
6.4	Recommendations for Future Work.....	157
	References	159
	Appendix A-Publications.....	163

List of Figures

FIGURE 1-1 JPEG COMPRESSION ATTACK	24
FIGURE 1-2 CONTRAST ADJUSTMENT ATTACK	24
FIGURE 1-3 REMOVAL ATTACK.....	25
FIGURE 1-4 CROPPING ATTACK.....	26
FIGURE 1-5 ADDITIVE NOISE ATTACK	26
FIGURE 1-6 AN EXAMPLE OF RESIZING ATTACKS	27
FIGURE 1-7 FILTERING ATTACK	27
FIGURE 1-8 PROCESSED IMAGES BY STIRMARK	29
FIGURE 2-1 MEAN SQUARE ERROR EXAMPLE BETWEEN TWO IMAGES	32
FIGURE 2-2 PEAK SIGNAL TO NOISE RATIO EXAMPLE BETWEEN TWO IMAGES.....	33
FIGURE 2-3 STRUCTURAL SIMILARITY INDEX MEASUREMENT EXAMPLE BETWEEN TWO IMAGES.....	34
FIGURE 2-4 NORMALIZED CORRELATION EXAMPLE BETWEEN TWO IMAGES.....	34
FIGURE 3-1 HOST COLOUR IMAGES WHICH ARE ALSO USED IN GREYSCALE FORM.....	47
FIGURE 3-2 BINARY WATERMARKS.....	47
FIGURE 3-3 A FLOW GRAPH FOR THE EMBEDDING PROCESS.	52
FIGURE 3-4 USED DCT COEFFICIENTS IN ONE 8×8 BLOCK OF THE HOST IMAGE.	52
FIGURE 3-5 A FLOW-GRAPH OF THE PROPOSED SHUFFLE SCHEME.....	54
FIGURE 3-6 A BLOCK DIAGRAM OF THE EXTRACTION PROCESS	55
FIGURE 3-7 THE AVERAGING AND CORRECTION PROCESS AFTER 50% HORIZONTAL CROPPING	56
FIGURE 3-8 THE AVERAGING AND CORRECTION PROCESS AFTER JPEG 50 COMPRESSION ATTACK	57
FIGURE 3-9 A FLOW GRAPH REPRESENTING THE DCS PROCESS	67
FIGURE 3-10 THE USED DCT COEFFICIENTS IN 8×8 SUB-BLOCK OF THE HOST IMAGE.	80
FIGURE 3-11 A FLOW GRAPH FOR THE EMBEDDING PROCESS.	80
FIGURE 3-12 A FLOW GRAPH FOR THE EXTRACTION PROCESS.	81
FIGURE 4-1 GRAPHICAL PRESENTATION FOR EMBEDDING STEPS.....	95
FIGURE 4-2 GRAPHICAL PRESENTATION FOR EXTRACTION STEPS.....	96
FIGURE 4-3 ANALYSIS BETWEEN GREY-SCALE IMAGES AND EACH OF R,G,B COMPONENTS.....	108
FIGURE 4-4 GRAPHICAL PRESENTATION FOR EMBEDDING STEPS.....	109
FIGURE 4-5 GRAPHICAL PRESENTATION FOR EXTRACTION STEPS.....	109
FIGURE 4-6 GRAPHICAL PRESENTATION FOR THE EMBEDDING STEPS.....	120

FIGURE 4-7 GRAPHICAL PRESENTATION FOR THE EXTRACTION STEPS.....	121
FIGURE 5-1 GRAPHICAL PRESENTATION FOR ERROR AND BINARY DECODERS	138
FIGURE 5-2 A FLOW GRAPH OF THE DCS PROCESS.....	140
FIGURE 5-3 GRAPHICAL PRESENTATION FOR EMBEDDING STEPS.....	141
FIGURE 5-4 GRAPHICAL PRESENTATION FOR EXTRACTION STEPS.....	142

List of Tables

TABLE 3-1 EFFECT OF USING SHUFFLE SCHEME AGAINST CROPPING ATTACK.....	54
TABLE 3-2 PSNR FOR DIFFERENT GREY-SCALE IMAGES WITH DIFFERENT EMBEDDING STRENGTHS AND WATERMARK SIZES	58
TABLE 3-3 SSIM FOR DIFFERENT GREY-SCALE IMAGES WITH DIFFERENT EMBEDDING STRENGTHS AND WATERMARK SIZES	59
TABLE 3-4 ORIGINAL AND WATERMARKED LENA IMAGES AT DIFFERENT EMBEDDING STRENGTHS	59
TABLE 3-5 NORMALIZED CORRELATION FOR LENA GREY-SCALE IMAGE	60
TABLE 3-6 WATERMARKED IMAGES AFTER ATTACKS	61
TABLE 3-7 EXTRACTED WATERMARKS AFTER ATTACKS AT WATERMARK EMBEDDING STRENGTH $\Delta = 14$	62
TABLE 3-8 EXTRACTED WATERMARKS AFTER ATTACKS AT WATERMARK EMBEDDING STRENGTH $\Delta = 16$	63
TABLE 3-9 MATLAB EXECUTION TIME	64
TABLE 3-10 PERFORMANCE EVALUATION AGAINST HIGH RESOLUTION IMAGES	65
TABLE 3-11 THE ORIGINAL DCS LOCATIONS BEFORE EMBEDDING THE WATERMARK.....	68
TABLE 3-12 THE NEW DCS LOCATIONS AFTER EMBEDDING THE WATERMARKS	69
TABLE 3-13 PSNR FOR DIFFERENT GREY-SCALE IMAGES WITH DIFFERENT EMBEDDING STRENGTHS AND WATERMARK SIZES	71
TABLE 3-14 SSIM FOR DIFFERENT GREY-SCALE IMAGES WITH DIFFERENT EMBEDDING STRENGTHS AND WATERMARK SIZES	71
TABLE 3-15 ORIGINAL AND WATERMARKED LENA IMAGES AT DIFFERENT EMBEDDING STRENGTHS	72
TABLE 3-16 NORMALIZED CORRELATION FOR LENA GREY-SCALE IMAGE.....	73
TABLE 3-17 WATERMARKED IMAGES AFTER ATTACKS	74

TABLE 3-18 EXTRACTED WATERMARKS AFTER ATTACKS AT WATERMARK EMBEDDING STRENGTH $\Delta =$ 14.....	75
TABLE 3-19 EXTRACTED WATERMARKS AFTER ATTACKS AT WATERMARK EMBEDDING STRENGTH $\Delta =$ 16.....	76
TABLE 3-20 MATLAB EXECUTION TIME	77
TABLE 3-21 PERFORMANCE EVALUATION AGAINST HIGH RESOLUTION IMAGES	78
TABLE 3-22 PSNR FOR DIFFERENT GREY-SCALE IMAGES WITH DIFFERENT EMBEDDING STRENGTHS AND WATERMARK SIZES	82
TABLE 3-23 SSIM FOR DIFFERENT GREY-SCALE IMAGES WITH DIFFERENT EMBEDDING STRENGTHS AND WATERMARK SIZES	83
TABLE 3-24 ORIGINAL AND WATERMARKED LENA IMAGES AT DIFFERENT EMBEDDING STRENGTHS	83
TABLE 3-25 NORMALIZED CORRELATION FOR LENA GREY-SCALE IMAGE.....	84
TABLE 3-26 WATERMARKED IMAGES AFTER ATTACKS.....	85
TABLE 3-27 EXTRACTED WATERMARKS AFTER ATTACKS AT WATERMARK EMBEDDING STRENGTH $\Delta =$ 14.....	86
TABLE 3-28 EXTRACTED WATERMARKS AFTER ATTACKS AT WATERMARK EMBEDDING STRENGTH $\Delta =$ 16.....	88
TABLE 3-29 MATLAB EXECUTION TIME	89
TABLE 3-30 PERFORMANCE EVALUATION AGAINST HIGH RESOLUTION IMAGES	90
TABLE 3-31 COMPARISON BETWEEN PROPOSED ALGORITHMS AND OTHERS	91
TABLE 3-32 COMPARISON OF ROBUSTNESS.....	91
TABLE 4-1 PSNR FOR WATERMARKED COLOUR IMAGES	97
TABLE 4-2 SSIM FOR WATERMARKED COLOUR IMAGES	97
TABLE 4-3 ORIGINAL AND WATERMARKED LENA IMAGES AT DIFFERENT EMBEDDING STRENGTHS	98
TABLE 4-4 NORMALIZED CORRELATION FOR LENA COLOUR IMAGE	99
TABLE 4-5 WATERMARKED IMAGES AFTER ATTACKS.....	100
TABLE 4-6 RECONSTRUCTED WATERMARKS AFTER ATTACKS AT $\Delta = 12$	101
TABLE 4-7 RECONSTRUCTED WATERMARKS AFTER ATTACKS AT $\Delta = 16$	102
TABLE 4-8 MATLAB EXECUTION TIME	103
TABLE 4-9 PERFORMANCE EVALUATION AGAINST HIGH RESOLUTION IMAGES	104
TABLE 4-10 PSNR FOR WATERMARKED COLOUR IMAGES	111
TABLE 4-11 SSIM FOR WATERMARKED COLOUR IMAGES	111
TABLE 4-12 ORIGINAL AND WATERMARKED LENA IMAGES AT DIFFERENT EMBEDDING STRENGTHS ..	112
TABLE 4-13 NORMALIZED CORRELATION FOR LENA COLOUR IMAGE	113

TABLE 4-14 WATERMARKED IMAGES AFTER ATTACKS	114
TABLE 4-15 RECONSTRUCTED WATERMARKS AFTER ATTACKS AT $\Delta = 24$	115
TABLE 4-16 RECONSTRUCTED WATERMARKS AFTER ATTACKS AT $\Delta = 34$	116
TABLE 4-17 MATLAB EXECUTION TIME	117
TABLE 4-18 PERFORMANCE EVALUATION AGAINST HIGH RESOLUTION IMAGES	118
TABLE 4-19 PSNR FOR WATERMARKED COLOUR IMAGES	122
TABLE 4-20 SSIM FOR WATERMARKED COLOUR IMAGES	123
TABLE 4-21 ORIGINAL AND WATERMARKED LENA IMAGES AT DIFFERENT EMBEDDING STRENGTHS ..	123
TABLE 4-22 NORMALIZED CORRELATION FOR THE LENA COLOUR IMAGE	124
TABLE 4-23 WATERMARKED IMAGES AFTER ATTACKS	125
TABLE 4-24 RECONSTRUCTED WATERMARKS AFTER ATTACKS AT $\Delta = 24$	126
TABLE 4-25 RECONSTRUCTED WATERMARKS AFTER ATTACKS AT $\Delta = 34$	128
TABLE 4-26 MATLAB EXECUTION TIME	130
TABLE 4-27 PERFORMANCE EVALUATION AGAINST HIGH RESOLUTION IMAGES	131
TABLE 4-28 COMPARISON BETWEEN THE PROPOSED ALGORITHMS AND OTHER BENCHMARK ALGORITHMS	132
TABLE 4-29 COMPARISON OF ROBUSTNESS	132
TABLE 4-30 COMPARISON OF ROBUSTNESS	133
TABLE 5-1 DCS LOCATIONS FOR ORIGINAL UN-WATERMARKED IMAGES	139
TABLE 5-2 DCS LOCATIONS FOR WATERMARKED IMAGES	139
TABLE 5-3 PSNR FOR DIFFERENT COLOUR IMAGES	143
TABLE 5-4 SSIM FOR DIFFERENT COLOUR IMAGES	143
TABLE 5-5 ORIGINAL AND WATERMARKED LENA IMAGES AT DIFFERENT EMBEDDING STRENGTHS	144
TABLE 5-6 NORMALIZED CORRELATION FOR LENA COLOUR IMAGE AT $\Delta = 24$	145
TABLE 5-7 NORMALIZED CORRELATION FOR LENA COLOUR IMAGE AT $\Delta = 34$	146
TABLE 5-8 MATLAB EXECUTION TIME	147
TABLE 5-9 PERFORMANCE EVALUATION AGAINST HIGH RESOLUTION IMAGES	148
TABLE 5-10 NORMALIZED CORRELATION FOR LENA COLOUR IMAGE AT $\Delta = 34$	149
TABLE 6-1 SAMPLES OF OTHER TESTED HOST AND WATERMARK IMAGES	152
TABLE 6-2 COMPARISON OF THE PROPOSED ALGORITHMS	155
TABLE 6-3 PERCEPTIBILITY VS. MAXIMUM WATERMARKING STRENGTHS	155
TABLE 6-4 NUMBER OF MULTIPLE EMBEDDING- EXTRACTION FOR EACH PROPOSED ALGORITHM	156

List of Abbreviations

BBP	Bit Per Pixel
CC	Correlation Coefficient
CD	Compact Disk
CONV	Convolution
dB	Decibel
DCB	Binary coded Decimal
DCS	DCT Coefficient Selection
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
FBI	Federal Bureau of Investigation
FFT	Fast Fourier Transform
H cropping	Horizontal cropping
HVS	Human Visual System
IDCT	Inverse Discrete Cosine Transform
JPEG	Joint Photographic Expert Group
LSB	Least Significant Bit
MMS	Multimedia Messaging Service
MSE	Mean Square Error
NC	Normalized Cross Correlation
OMAP	Open Multimedia Application Platform
PDA	Personal Digital Assistant
PSNR	Peak Signal to Noise Ratio
RESC	Rescaling
RML	Remove line
ROT	Rotation
ROTCROP	Rotation with Cropping
ROTSKALE	Rotation with Scaling
SS	Structural Similarity

SSIM	Structural Similarity Index Measurement
USB	Universal Serial Bus
USC-SIPI	University of South California - Signal & Image Processing Institute
V cropping	Vertical cropping

List of Variables

B_n	number of watermark bits that can be embedded
C	Contrast
ir	red component of the original image
ig	green component of the original image
ib	blue component of the original image
i	original image
wr	red component of the watermarked image
wg	green component of the watermarked image
wb	blue component of the watermarked image
Δ	watermark embedding strength
N_{HB}	sub-blocks of the host image
N_{wB}	number of watermark blocks
n	number of watermark copies
Z_w	size of the watermark
Z_h	size of the host image
U_c	eight DCT coefficients inside each sub-block
Q_e	quantization to even number
Q_o	Quantization to odd number
w_{SB}	Watermark shuffled bits
w	watermarked image
e	extracted watermark

<i>L</i>	Luminance
<i>S</i>	Structure
<i>XY</i>	rows and columns of digital image

Chapter 1 Introduction

1.1 The Importance of Digital Image Watermarking

Over the past few years, there has been rapid growth in computer networks and more specifically, the World Wide Web. This fact, coupled with the exponential increase of computer performance and powerful image processing software, has facilitated the distribution and made it much easier to make unlimited number of copies of an original image. Images can be manipulated or modified easily with a wide range of software packages and people often claim that these modified images are theirs when in fact somebody else originally produced them [1, 2]. The simplicity with which such images can be duplicated and mutated has created the need for efficient copyright protection methods. Digital watermarks have been proposed as a way to tackle this continuing issue. Digital watermarking is a technology for embedding various types of information in digital content [3]. This digital signature could discourage copyright violation, and may help determine the authenticity and ownership of an image.

Commercial organizations such as Digimarc, AquaMobile and MarkAny which service a broad range of industries around the world have joined the digital watermarking alliance (DWA) [4]. DWA is actively involved in the commercialization of digital watermarking-based applications, systems and services. The DWA also delivers a broad range of watermarking solutions to customers around the world [4]. Digimarc image online solution is one example of commercial applications built for copyright protection [5].

Applications that require still image watermarking include: images captured by digital cameras and mobile phones, medical images, satellite images and artistic images. For example, many mobile phones are equipped with high resolution digital cameras. Their users can capture images and share them with others by sending them as email attachments, multimedia messages or via Bluetooth. Digital watermarking provides the security of knowing that no matter how or where images appear they

carry the notice of ownership. This can be done by embedding the phone number including the international calling code onto the images.

This project will concentrate on copyright protection of still images. In digital image watermarking, copyright defending information is embedded in the image in the form of a watermark. The original image must not be affected i.e., visibly degraded by the presence of this watermark. Another main prerequisite for copyright protection applications is the robustness of the watermark. Thus, the watermark must withstand unauthorized detection and decoding. In addition, the watermark must survive the normal image processing techniques (e.g. compression), as well as intentional attacks (attempts to destroy or remove the watermark).

1.2 Aims and Contributions

Digital watermarking Technology is used widely to protect copyrights of images and to aid owners in asserting their intellectual property rights of the works of art they create. The basic components of any watermarking technique consist of an embedding algorithm that inserts information, the watermark, and an extraction algorithm that defines and tests an image to see if a particular watermark is contained in the image. The objective of this research is to develop image watermarking techniques which can be used for copyright protection of images captured by digital cameras or mobile phone cameras. Each watermarking application has its own specific requirements. The developed techniques here are well-designed with unique features to integrate the major watermarking requirements. The proposed techniques can support different applications because they allow the users to adjust the strength of the watermark. The proposed techniques have the following features:

1. **Visually recognizable extraction** – The viewer can evaluate the results subjectively and objectively. Handwritten signatures and mobile phone numbers will be used as watermarks rather than the conventional pseudo random numbers.
2. **Invisibility or transparency** – the watermark is not visible in the image under typical viewing conditions
3. **Robustness** to attacks – the watermark can still be extracted and made difficult for an attacker to remove even after the image has gone under

different attacks and geometric distortion. A shuffle scheme will be applied to shield the proposed algorithms against cropping attacks.

4. **Capacity**–the watermarking technique will be capable of hiding up to 25% of the host image size.
5. **Security**–A DCT coefficient(s) selection process (DCS) will be developed to increase the security of the proposed algorithms.
6. **Blind**– the watermark can be recovered without any reference to the original host image.
7. **Error reduction** process will be applied after the reconstruction of the watermark.

1.3 Applications of Watermarking

Copy protection and copyright protection for digital data can be achieved using encryption and watermarking [3]. Encryption techniques are used to protect digital data during the transmission process [6]. After the receiver has received and decrypted the data, the retrieved data should be identical to the original data. Watermarking techniques can compliment encryption by embedding a secret imperceptible signal (a watermark) directly into the original data in such a way that it always remains present. This watermark can be used in a wide variety of applications as described in the proceeding sections. More details of these applications can be found in [2, 6].

1.3.1 Copyright Owner Identification

For the protection of intellectual property, the data owner can embed a watermark representing copyright information in his/her data. This watermark can prove the ownership in court when someone has infringed on the copyright. Digimarc's watermark for images [7] was designed for this type of application. It detects a watermark; it contacts a central database over the Internet and uses the watermark message as a key to find contact information for the image's owner.

1.3.2 Fingerprinting or Transaction Tracking

Fingerprinting techniques can be used to trace the source of illegal copies [6]. In this case, the owner can embed different watermarks in the copies of the data that are supplied to different customers. Fingerprinting can be compared to embedding a serial number that is related to the customer's identity in the data. It enables the intellectual property owner to identify customers who have breached their license agreement by supplying the data to third parties [6]. The person responsible for misuse is referred to as *traitor* but the person who receives the work from a traitor is a *pirate*. Anyone who tries to remove or forge a watermark is called an *adversary*.

1.3.3 Broadcast Monitoring

This is used to identify when and where the works are broadcast by recognizing watermarks that are embedded in them. By embedding watermarks in commercial advertisements, an automated monitoring system can verify whether advertisements are broadcasted as contracted [6]. There are many organizations interested in monitoring broadcasting. They want to make sure that they receive all the airtime they purchase from broadcasters. Techniques for checking that can be classified into two types: passive monitoring that tries to directly recognize the content being broadcast, and active monitoring which relies on the associated information that is broadcasted along with the content.

1.3.4 Data Hiding

Watermarking techniques can be used for the transmission of secret private messages. Since various governments restrict the use of encryption services, people may hide their messages in other data [6].

1.3.5 Data Authentication

Fragile watermarks [8] can be used to check the authenticity of the data. A fragile watermark indicates whether the data has been altered and supplies localization information as to where the data was altered. This can be useful in legal investigations, for example.

1.3.6 Copy Control

The watermark can be used to stop recording equipments from copying copyrighted contents. The information stored in a watermark can directly control digital recording devices for copy protection purposes [6]. In other words, a watermark detector will be fitted in every device, so every time *never-copy* is detected, it will not allow recording. The goal of using this kind of application is to ban people from making illegal copies of copyrighted content.

1.3.7 Device Control

This involves using watermarks to make a device react to specific contents. There are many applications in which devices react to watermarks they detect in order to add value to content rather than restrict the usage. For example, when a Digimarc MediaBridge digital watermark on product packaging is read by a consumer's PC camera, scanner or other optical device, special watermark-reading software initiates the display of a web destination or a web-based application specified by the company or targeted to a specific customer.

1.4 Watermarking Classifications

Watermarking techniques have various kinds of classifications depending on the nature of its application [2]. Each watermarking application has its own specific requirements. Therefore, there is no set of requirements to be met by all watermarking techniques. Nevertheless, some general directions can be given for most of the applications mentioned above [6].

1.4.1 Invisible versus Visible Watermarking

The watermarking here is categorized based on visual appearance. If the watermark is visible to the observer's eye and shows a type of information like a trademark or any other necessary information, then it is called a *visible watermark*. In contrast, if the watermark is embedded in a way that will make it invisible and undetectable by observers, then this is called an *invisible watermark*. Invisible watermarks have an advantage over visible watermarks in that their location may be unknown. A common practice is to distribute the watermark (or watermarks) across the entire image [1].

1.4.2 Blind versus Non-Blind Watermarking

This type of watermarking is based on whether the original image is required for the recovery system or not. If the original image is not needed, then the method is called *blind watermarking* (also referred to as complete). Otherwise, if the original image is needed, then it is *non-blind watermarking* (also referred to as incomplete). In some applications, like copyright protection and data monitoring, watermark extraction algorithms can use the original un-watermarked data to find the watermark [9]. In other applications e.g., copy protection, the watermark extraction algorithm does not have access to the original un-watermarked data. This renders the watermark extraction more difficult.

1.4.3 Robust versus Fragile Watermarking

Efficient watermarking systems have to be *robust* against different types of attacks. In other words, it should be difficult to remove the embedded watermark. The other type of watermarking is *fragile watermarking*. A fragile watermark that has to prove the authenticity of the host data does not have to be robust against image processing techniques or intentional alterations of the host data, since failure to detect the watermark proves that the host data has been modified and is no longer authentic [6]. Fragile watermarking is used to detect the location of the changes in the watermarked data. It has the ability to define the modified area of the watermarked data.

1.5 Watermarking Requirements

Each watermarking application has its own requirements. All of these watermarking requirements are related to each other. There are many requirements for a well-designed watermark. Some of these requirements are described in what follows. More details can be found in [2, 6, 10].

1.5.1 Perceptual Transparency or Imperceptibility

In most applications the watermarking algorithm must embed the watermark without affecting the quality of the underlying host data. The embedded watermark is *imperceptible* if humans cannot distinguish the original data from the data with the inserted watermark [6].

1.5.2 Payload

The *payload* is the amount of information that can be stored in a watermark. This depends on the application. For example, for copy control applications 1-bit is sufficient to indicate to the device whether it is allowed to copy the work or not. For example, the protection of intellectual property rights, require more bits [6, 9] of information to be embedded in the host data .

1.5.3 Robustness

Robustness describes how well a watermark survives common image processing operations. In some applications, it is desirable for the watermark to stay almost intact in the host data, even if the quality of the host data is degraded intentionally or unintentionally. An example of unintentional degradation is the application of lossy compression techniques to reduce bit rates and increase efficiency [11]. Other unintentional degrading processing techniques include filtering, re-sampling, analogue-to-digital (A/D) and D/A conversions. As a result, there should be no way in which the watermark can be removed or modified without sufficient degradation of the perceptual quality of the host data [9].

1.5.4 Security

Watermarking security is the ability of a watermark to resist hostile attacks. The security of watermarking techniques can be interpreted in the same way as the security of encryption techniques [6]. A watermarking technique is truly secure if knowing the exact algorithms for embedding and extracting the watermark does not help an unauthorized party detect the presence of the watermark or remove it. This can be achieved by the choice of one or more secret keys [6].

1.5.5 Trustworthiness

A satisfactory watermarking scheme should also guarantee that it is impossible to generate counterfeit watermarks and should provide trustworthy evidence to protect the rightful ownership [10, 12].

1.6 Some Significant Known Attacks

Watermark robustness under modification is an essential issue for copyright protection [13, 14]. Any provider or user can modify an original digital image to improve quality, compress data, and so on. Protecting copyrights while maintaining sufficient quality under these conditions is desirable [13]. There are a number of well known attacks that are carried out on watermarking systems [15]. There are also powerful tools such as StirMark and unZign, which are used to generate watermarking attacks. Examples of some image processing attacks are described below.

1.6.1 JPEG Compression Attack

Image compression is used to reduce the data-content size of a digital image. Image compression works by eliminating the redundancies of information, thus keeping the essential information in an image. In general, image compression techniques can be grouped in two categories: *lossless* and *lossy* methods. Lossless image compression techniques are used to reduce the data size of image files while maintaining the original image quality. Lossy image compression techniques, on the other hand, achieve greater compression ratios but they cause some degradation to the original image quality. A number of standardized image compression techniques have been developed to support the requirements of various industries. The most common image data compression standard is the JPEG standard [16]. JPEG stands for the *Joint Photographic Experts Group* (JPEG) standard. The JPEG standard handles grey-scale and colour images of different resolutions. It can support many industries that need to transport and archive images. It is used in many applications like graphic art, desktop publishing and medical imaging [16].

In digital images, the original source material may be compressed for more efficient storage or transmission. Therefore, it is important to examine whether or not the proposed watermarking algorithms can survive JPEG compression. The quality rates for JPEG compression can be set to different values. Higher compression ratios yield coarse quantisation for DCT coefficients. Hence, the watermark will be destroyed and become indiscernible. However, in this situation, the quality of the JPEG compressed image (without being watermarked) will be degraded severely so that the processes of digital watermarking become less meaningful. Figure 1.1 illustrates

the original Lena image that has been compressed to different quality rates. The quality rates range between 0 and 100; higher numbers mean higher quality (less image degradation due to compression), but the resulting file size is expected to be larger.

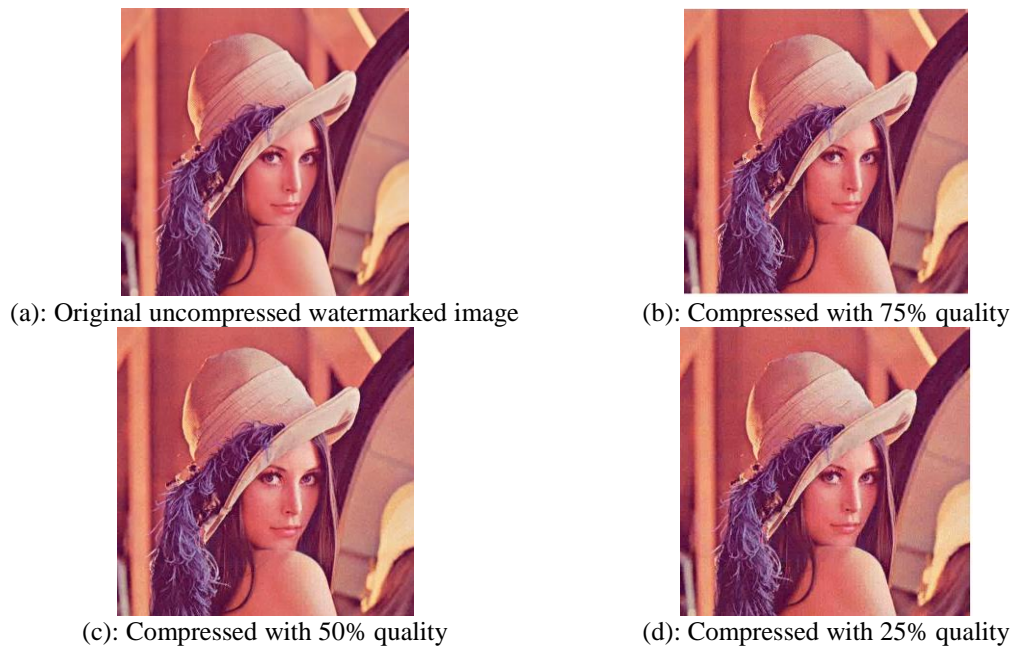


Figure 1-1 JPEG compression attack

1.6.2 Image Enhancement Operations

Digital cameras have been widely used to capture images in digital format. As a result, captured images can be more easily processed. The contrast of an image is usually adjusted to enhance the subjective quality. Image quality of different contrast enhancement are shown in Figures 1.2(b) and Figure 1.2(c).



Figure 1-2 Contrast adjustment attack

1.6.3 Removal Attack

Removal attacks aim at the complete removal of the watermark information from the watermarked data. Removal attacks try to severely impair the embedded watermark while maintaining the quality of the attacked image.



(a): Original image

(b): 3×3 Wiener filtered image

(c): 5×5 Wiener filtered image

Figure 1-3 Removal attack

1.6.4 Cropping Attack

During image manipulation, parts of the image could be cropped. A pirate could try to remove the watermark by cutting some parts of the image. It would be interesting to examine the watermarked images against cropping, where the spatial information are discarded. For simplicity, the missing portions are filled with zero values. This distributes more noise over the entire results and influences the visual recognition. The watermark should cover the entire image so it will be robust to cropping [13]. An example of the cropping attack is shown in Figure 1.4.



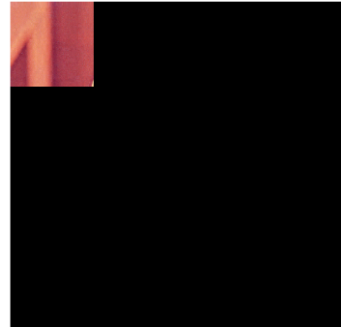
(a): original un-cropped image



(b): Cropped side to 75% Vertically



(c): Cropped side to 75% Horizontally



(d): Cropped both sides to 75%

Figure 1-4 Cropping attack

1.6.5 Additive Noise

This can result from certain applications such as the use of A/D and D/A converters or from transmission errors. Authors often claim that their copyright marking techniques survive this kind of attack, but many forget to mention the maximum level of acceptable noise that can be handled by these techniques [9]. An example of additive noise attack is shown in Figure 1.5.



(a): Original Lena image



(b): Gaussian noise



(c): Salt & pepper noise

Figure 1-5 Additive noise attack

1.6.6 Resize Attack

The accurate detection of watermarks in geometrically modified images is a difficult task [13]. Geometric attacks do not actually remove the embedded watermark itself, but aim to change the synchronization of the embedded information. The detector would recover the embedded watermark information when perfect synchronization is regained. To test the robustness of the proposed algorithms against resizing attacks, the original Lena image has been resized to different scales. Note that the attacked image must be restored to its original dimensions before extracting the watermark. The image with original size of 512×512 is shown in Figure 1.6(a). Figure 1.6(b) is

the Lena image resized to 256×256 . In Figure 1.6(c) the Lena image is resized to 128×128 .

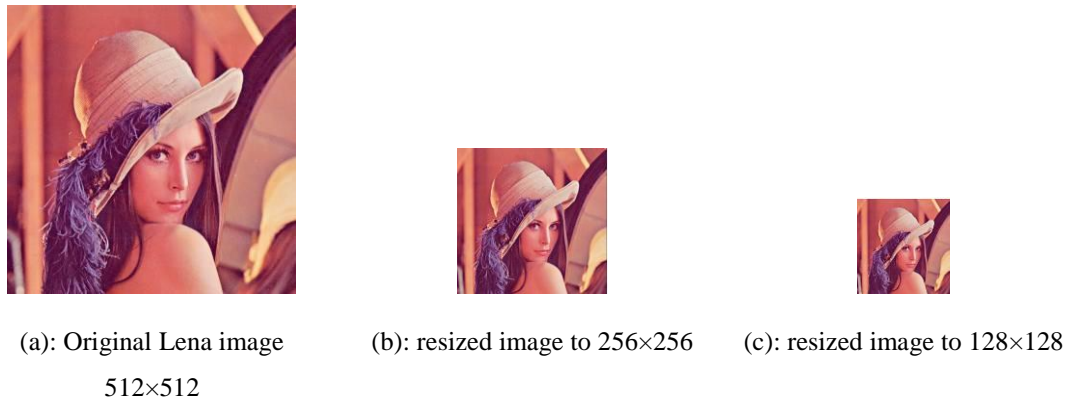


Figure 1-6 An example of resizing attacks

1.6.7 Filtering

The robustness of watermarking algorithms is usually examined against low-pass and median filters. The watermarked images may still be recognizable when undergoing filtering levels of 3×3 mask size, but higher levels of filtering using 5×5 mask size, would spoil the quality of the watermarked image which by turn would spoil the watermark. An example of the filtering attack is shown in Figure 1.7.

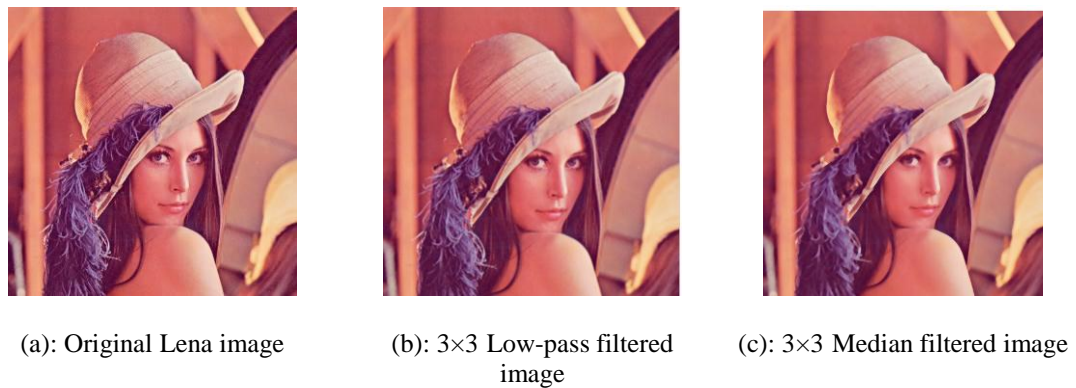


Figure 1-7 Filtering attack

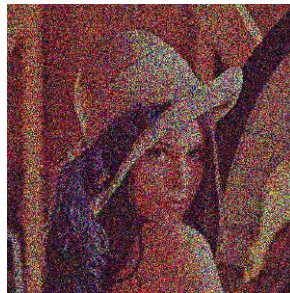
1.6.8 Standard Assessment Tools

Benchmarking tools are used to evaluate the robustness of a watermarking technique against attacks. Several tools are popular in the market such as, Checkmark, Optimark, and Stirmark. Checkmark was developed by Shelby Pereira [17]. It is a benchmarking tool for digital watermarking. It can run on Matlab under UNIX and

Windows. Optimark is another benchmarking tool for still image watermarking algorithms which was developed in the Artificial Intelligence and Information Analysis Laboratory at the Department of Informatics, Aristotle University of Thessaloniki, Greece [18]. In November 1997, the first version of Stirmark was introduced as a tool for robustness testing of image watermarking algorithms [19]. Stirmark has been developed by Fabien Petitcolas during his Ph.D. at Cambridge University, UK. Stirmark has gained large interest from the watermarking community and it is currently the most widely used benchmarking suite for digital watermarking technologies. Given a watermarked input image, Stirmark generates a number of modified images (attacked images) which can then be used to verify the performance and test if the embedded watermark can still be extracted. In the following Table some examples are given for several images processed by Stirmark.



(a): Original watermarked image



(b): Stirmark_NOISE_20



(c): Stirmark_ROT_-2



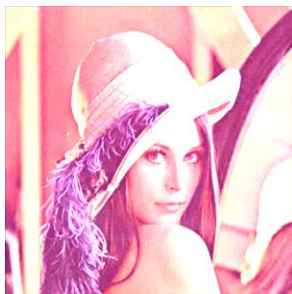
(d): Stirmark_ROTSCALE_0.5



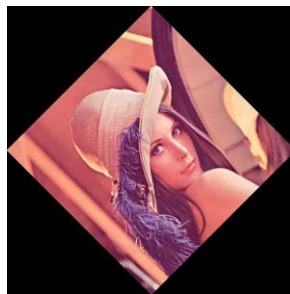
(e): Stirmark_SS_1



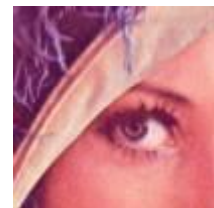
(f): Stirmark_AFFINE_4



(g): Stirmark_CONV_1



(h): Stirmark_ROT_45



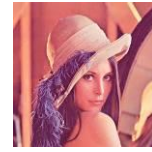
(i): Stirmark_CROP_20



(j): Stirmark_RML_100



(k): Stirmark_ROTROP_-0.5



(l): Stirmark_RESC_50

Figure 1-8 Processed images by Stirmark

1.7 Challenges in Digital Watermarking

Digital watermarking has recently become an important field of research. Many researchers produced papers covering digital watermarking techniques, attacks, applications and analysis. Digital watermarking challenges include design considerations, requirements, robustness, tradeoffs involved and speed. One of the major digital watermarking challenges is to design an embedding-extraction system that would not affect the quality of the image while at the same time, would satisfy the conditions of security and high robustness. The efficient watermarking system must fulfill the common requirements of transparency and robustness. A new challenge arises when a speedy embedding technique is needed so that users do not face unacceptable delays before they download their marked content. Another challenge is to make the watermarking system deal easily with different file formats, different colour formats, different sizes and types of images, different sizes and types of watermarks and different applications. Data capacity which is defined as how much data can be added before disturbing the quality of the image is yet another challenge for digital watermarking systems. The capability of any watermarking system to hide large or small amounts of data while maintaining the robustness and quality is also an important challenge in watermarking.

1.8 Overview of the Thesis

Chapter two describes the performance metrics and measurements used in watermarking. It also describes the state-of-the-art in watermarking. Chapter three investigates the performance of the developed grey-scale watermarking techniques that works in the Discrete Cosine Transform (DCT). In this context, the chapter considers the shuffle scheme and its effect on the developed techniques. This chapter also comprises three watermarking algorithms for grey-scale images. Chapter four considers three colour watermarking algorithms, the first algorithm introduces a colour watermarking technique using YCbCr Model, the second watermarking algorithm is based on the green channel of the RGB model, while the third algorithm is a high capacity watermarking algorithm. Chapter five discusses a watermarking algorithm for digital images captured by mobile phone cameras. Chapter six represents the conclusions and a plan for the future work. Appendix A includes a list of the publications produced from this work.

Chapter 2 Digital Image Watermarking

2.1 Overview

This chapter introduces digital image watermarking performance metrics and measurements. Also, it covers a literature survey of watermarking, limitations in the current state of art and knowledge gaps.

2.2 Evaluation and Benchmarking of Watermarking Systems

When designing digital watermarking methods, it is important to address proper evaluation and benchmarking. This is required to evaluate the robustness and the perceptual distortion introduced through the watermarking process. In general, for good benchmarking and performance evaluation one has to ensure that the methods under investigation are tested under comparable conditions [9]. The robustness of watermarks depends on the following aspects [9]:

- *Amount of embedded information*: it is a significant parameter since it directly influences the watermark robustness. The more information is embedded, the lower is the watermark robustness.
- *Watermark embedding strength*: there is a trade-off between the watermark embedding strength and the watermark perceptibility. To increase robustness, stronger embedding is required but that will increase the perceptibility of the watermark.
- *Size and nature of host data*: the size of data has a direct impact on the robustness of the embedded watermark.

Taking the above parameters into account, it is realized that for fair benchmarking and performance evaluation, watermarking methods need to be tested on different data sets. Also, in order to compute statistically valid results, the methods have to be evaluated using many different keys and varying watermarks [9]. The amount of embedded information is usually fixed and depends on the application.

2.2.1 Watermarking Datasets

It is important to test watermarking algorithms on many different images and for fair comparison the same set of sample images should be used. Some image datasets already exist for image processing research. The USC-SIPI image datasets [20, 21] is an example of such datasets where one can find the classics; Lena, Baboon, Peppers, etc. Kodak images are another dataset of standard images [22]. Kodak images are also used widely to compare results in watermarking techniques, both visually and quantitatively [22].

2.2.2 Performance Evaluation and Representation

This section lists a number of metrics that quantify image degradation. These metrics have been applied widely for image quality assessment. The metrics measure quality degradation using pixel-based comparisons [23]. The Mean Square Error (*MSE*) compares two images on a pixel-by-pixel basis. Mathematically, *MSE* is expressed as [9, 23]:

$$\begin{aligned} \text{For grey-scale images: } \quad MSE_{-grey} &= \frac{1}{xy} \sum_{x,y} (i_{x,y} - w_{x,y})^2 \\ \text{For colour images:} & \\ MSE_{-colour} &= \frac{1}{3xy} \sum_{x,y} (ir_{x,y} - wr_{x,y})^2 + (ig_{x,y} - wg_{x,y})^2 + (ib_{x,y} - wb_{x,y})^2 \end{aligned} \quad (2.1)$$

Where *ir*, *ig* & *ib* are the R,G and B components of the original colour image and *wr*, *wg* & *wb* are the R,G and B components of the watermarked colour image. This measure gives an indication of how much degradation was introduced at a pixel based level. The higher the *MSE*, the greater the level of degradation. Figure 2.1 shows an *MSE* example when applied to the original Lena and a watermarked version with *MSE*=44.66.

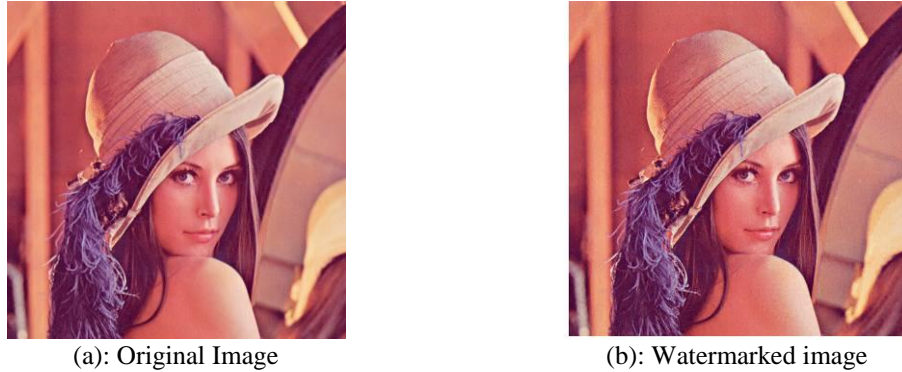


Figure 2-1 Mean Square Error example between two images

The peak signal to noise ratio (*PSNR*) is a commonly used image quality metric. *PSNR* is given by [9, 23]:

$$PSNR = 10 \cdot \log_{10}(255^2 / MSE) \quad (2.2)$$

Where i, w is the original image and watermarked image respectively and x, y are the image pixels. Thus, two images that are exactly the same will produce an infinite *PSNR* value. By using the watermarking algorithm in [24] an example of *PSNR* computation between the original Lena host image and the watermarked version with $PSNR=50.9684$ is shown in Figure 2.2.

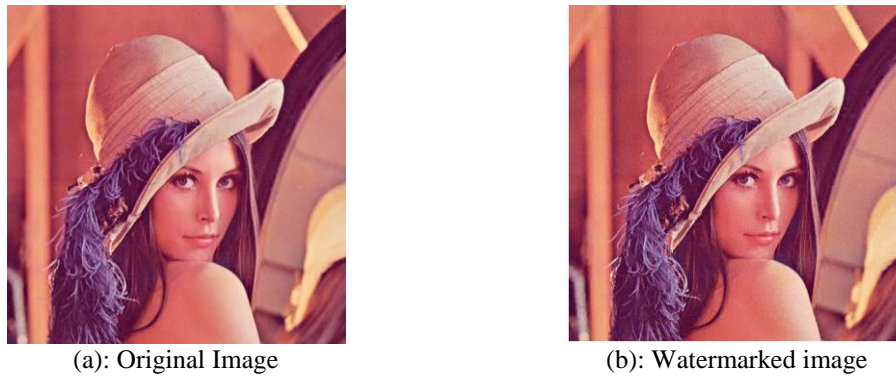


Figure 2-2 Peak Signal to Noise Ratio example between two images

The Structural Similarity Index Measurement *SSIM* is a measure that compares local pattern of pixels intensities that have been normalized for luminance and contrast [25]. The higher *SSIM* is, the larger the similarity between the compared images. The *SSIM* range between zero and one. *SSIM* is expressed as [25]:

$$SSIM(i, w) = [L(i, w)]^\alpha \cdot [C(i, w)]^\beta \cdot [S(i, w)]^\gamma \quad (2.3)$$

Where i, w represent the original and watermarked image respectively and L, C and S represent the luminance, contrast and the structure. α, β, γ are parameters used to adjust the relative importance of the luminance, contrast and structure components [25]. The *SSIM* index is a full reference metric, in other words, the measuring of image quality based on an initial uncompressed or distortion-free image as reference. *SSIM* is designed to improve on traditional methods like peak signal-to-noise ratio (*PSNR*) and mean squared error (*MSE*), which have proved to be inconsistent with human eye perception [25]. An example of *SSIM* evaluation is illustrated in Figure 2.3 where the *SSIM* value of the watermarked image is equal to 0.9631.

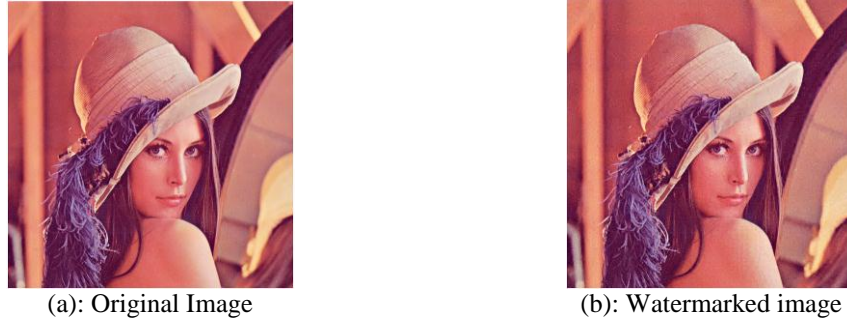


Figure 2-3 Structural Similarity Index Measurement example between two images

Another performance measure for image watermarking schemes is the normalized cross correlation NC , and it can be defined as [9, 23]:

$$NC = \frac{\sum_{x,y} i_{x,y} e_{x,y}}{\sqrt{\sum_{x,y} i_{x,y}^2 \sum_{x,y} e_{x,y}^2}} \quad (2.4)$$

Where i, e are the original watermark and the extracted watermark respectively. Figure 2.4 shows an example of Normalized Correlation NC computation between an original watermark and a reconstructed watermark with $NC=0.9113$

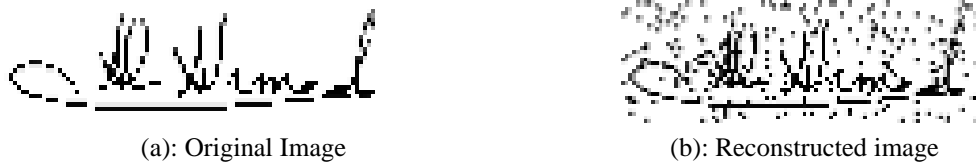


Figure 2-4 Normalized Correlation example between two images

2.3 Literature Survey of Watermarking Techniques

There are two basic categories for image watermark encoding: spatial-domain techniques and transform-domain techniques. This section first describes several spatial watermarking algorithms [26-33]. Many of the spatial watermarking techniques provide simple and effective schemes for embedding an invisible watermark into the original image but are not robust to common image alterations. In the simplest sense, these are the first perceptually based watermarking techniques that rely on a scheme for watermark encoding which will produce resulting images of high quality but not necessarily robust to attacks [26-33]. Another way to mark an image is to transform it into another domain by using Fourier, DCT, wavelet, etc. before marking it [12, 14, 24, 34-46]. Due to the complicated calculations of forward and inverse transform, these methods generally are more complex and involve higher computational costs than spatial domain methods. The watermark is incorporated

directly into the transform coefficients of an image. The inverse-transformed coefficients form the marked image. These types of algorithms are often called spectral watermarks, and commonly use the frequency sensitivity characteristics of the human visual system to ensure that the watermark is invisible. Many of these techniques are non-blind watermarks that require the original image to verify the mark. As illustrated before, algorithms that do not require the original image for testing are called blind watermarks.

2.3.1 Spatial Domain Techniques

The spatial-based methods embed a watermark by directly modifying the pixel values of the host image. A commonly used method in spatial domain is the least-significant-bit (LSB) method [26, 27], which is fragile to any possible attacks. Improved methods were proposed against image compression and other image manipulations.

In [27] a watermark has been generated using m-sequence generator. The watermark was either embedded or added to the least significant bits of the original image to produce the watermarked image. The resulting image contained an invisible watermark with simple extraction procedure. The watermark, however, was not robust to additive noise.

In [28] the authors affirmed that the LSB algorithms can be attacked in various ways, by simply changing the LSB's of the host image, which changes the embedded watermark and could destroy it. In such a case the receiver of the message has no way to tell that an attack took place. In [28] another algorithm has been developed that uses the LSB to hide a secret message and an error correction code is used to increase the probability of retrieving the secret message. If the watermarked image undergoes a slightly simple modification, then the receiver will know if there was an attack during or after transmission.

In [29] the saturation component of the HIS colour model is adopted to hide the watermark into DC components of the colour image directly in the spatial domain, Hiding a watermark in the saturation component only changes the amount of white light mixed with hue. Since the colour of a watermarked image is slightly brighter or

darker than the original image, it is less sensitive to human visual systems than changes of hue.

The authors in [30] proposed a new spatial domain probability based watermarking scheme for colour Images. The blue channel of the colour image has been used for watermark embedding. A Host image is divided into 8×8 blocks and each bit of the binary encoded watermark is embedded in each block. For each inserted bit, the intensities of all the pixels in the block are modified according to the embedding algorithm. In this method the number of the total bits of the watermark must be less than or equal to half the total number of 8×8 blocks to reduce the distortion to the host image. The proposed method is quite robust against some common image processing operations, such as filtering, scaling and rotation. However, the watermark bits are embedded into the whole image which makes it less robust to cropping attacks since some data would be lost in cropping.

In [31] another spatial domain transform for colour images was introduced where the watermark was embedded four times in different positions. Each bit of the binary encoded watermark is embedded by modifying the intensities of an 8×8 non overlapping block of the blue component of the host image.

The authors in [30, 31] have not given any justification for the choice of the blue channel and hence the obtained invisibility metrics are not very good. Furthermore, using blue channel for watermark embedding contradicts the results and experimental conclusion in [35] where it is stated that the choice of the green channel in RGB format gives the best invisibility and robustness against many attacks.

The authors in [37] proposed an adaptive digital image watermarking technique. In this technique the watermark is a visually recognizable binary image rather than a randomly generated sequence of bits. To prevent un-authorized access, the watermark is first permuted into scrambled data. The watermark is then embedded by modifying the intensities of some selected pixels in such a way that the modification is not noticeable to the human eye. Robustness has been evaluated

against low-pass filter and jpeg compression only. Moreover, the technique is non-blind.

In [32] another spatial domain, non blind algorithm called YSCE for embedding coloured images is proposed. The 24 bits/pixels RGB images are converted to ITU-R601 standard which is known as YCrCb, where the watermark is embedded in the Y layer. The algorithm manages to survive several attacks such as: JPEG, filtering, cropping and noise addition.

A modified version of the technique in [32] is presented in [33] for processing colour images. This algorithm depends on choosing the colour layer with maximum edge energy for embedding the watermark. The watermark pixels are scrambled together with the blocks of the selected colour using two different keys which gives double uncertainty that increases the robustness. The embedding process is adaptive and it takes into consideration the nature of the images and the human visual system. The method is robust and it manages to survive several intentional and unintentional attacks. The robustness are measured using the mean absolute error MAE; which is rarely used to evaluate the robustness in watermarking algorithms. The technique is non-blind since it requires the original host for extraction. The authors have not compared the results with spatial domain watermarking techniques, and the invisibility qualities results have been compared with a frequency domain developed techniques using DWT.

2.3.2 Frequency Domain Techniques

In contrast to spatial domain methods, frequency domain methods, such as DCT and DWT, modify the transform coefficients. A combined DWT-DCT has been implemented in [47]. The watermark was embedded by applying DCT on the selected DWT sub-bands. Combining the two transforms has enhanced the watermark performance in comparison with DWT only. However, the experimental results compared DWT-DCT and DWT only and the obtained results were not compared with DCT-based techniques. The developed DWT-DCT combination dealt only with grey-scale images.

Authors in [48] proposed a DCT domain watermarking algorithm. The embedded information is Arnold transformed and this will break the correlation between the

watermark data and the watermarked image. Grey-scale host images of size 256×256 and watermark size of 50×50 was used. The experimental results show that the method was robust to JPEG compression and additive noise. However the invisibility qualities were not mentioned or evaluated by any known method and the robustness against the mentioned attacks is evaluated subjectively without any objective measurements.

A digital watermarking algorithm is proposed in [49] using the DC components of the 8×8 DCT blocks. Comparison between the proposed method in DCT and its spatial domain was given. The robustness of the method was evaluated using additive noise only. No other attacks were mentioned or tested. The sizes of the host images and watermarks were not given.

Another watermarking algorithm for image authentication is proposed in [50]. The original image is subdivided into 8×8 blocks and each block is transformed to DCT domain. The embedded information was hidden in the middle frequency coefficients. The experimental results show that the invisibility qualities are moderate since PSNR for some shown images were 35.3. The robustness was evaluated against jpeg compression only. The embedded watermark is detected and not extracted. A possibility of false detection might occur.

A blind high capacity watermarking algorithm in the DCT domain is proposed in [51]. The coefficients of the DCT sub-blocks were scanned in zigzag manner. Blocks of 64×64 and 16×16 were used for the original image and watermark, respectively. Each sub-block of the watermark is embedded in a sub-block of the original image. Grey-scale Lena image of size 512×512 was used in the experimental tests with different watermark sizes. The invisibility quality of the method at watermark size of 128×128 was measured using PSNR and it was found to be 31.5.

A non-blind colour DCT-based watermarking algorithm is proposed in [52]. The watermarking image was a 24-bit colour image with a size of 64×64 pixels. The colour watermark was divided into its original R,G and B components. Binary watermarking sequences were generated from each component. The generated binary sequences from each watermark channel were embedded in the R,G, and B components. The zigzag order was applied for each 8×8 sub-blocks of the DCT. The

coefficients with order numbers from 3-10 were selected to be embedded for watermarking. The experimental results were evaluated by using PSNR, NC and subjective judgment. The robustness was examined using additive noise and jpeg compression only.

Another blind block based DCT watermarking technique for grey-scale images using one dimensional Walsh coding is presented in [53]. The 1-D Walsh code is applied to the handwritten signatures before embedding it. The 1-D Walsh code increased the used signature size from 64×64 to 64×256 . The embedding process is achieved by placing the encoded signature's bits into the low frequency coefficients of the DCT Blocks. A number of 512×512 grey-scale images have been used in the test. The robustness was evaluated against jpeg only. The disadvantage of Walsh coding is that it increases the size of the embedded watermarks, which in turn influences the invisibility qualities of the watermarking algorithms.

Another DCT watermarking method is proposed in [54]. Arnold transform has been used to scramble the watermark. The zigzag process has been applied for each 8×8 sub-blocks of the DCT. The watermark was embedded in the low-mid frequency AC coefficients. Grey-scale host images of size 512×512 were used. Two watermarks of size 32×32 were also used. The invisibility qualities were evaluated using PSNR. The average PSNR for the test images was found to be 32.3.

A watermarking technique is proposed in [55]. The AC coefficients of the host image in the DCT domain were modified to embed the watermark. Grey-scale test images of size 512×512 and a 128×128 watermark were used. The average PSNR was 41. The method was tested against Low-pass, high-pass and JPEG attacks only. The proposed method tried to tackle the speed of embedding and extraction of the watermark. However, the measured time for embedding and extraction was very high.

A blind digital image watermarking algorithm in the DCT domain is presented in [56]. The average value of the DCT coefficients was used as a threshold to realize the watermark embedding. The watermark was embedded in the low frequency components of the DCT. Before the watermark embedding the DCT blocks were transformed into one dimensional data in the form of zigzag. Gray-scale images of

size 512×512 and a watermark of size 32×32 were used. The invisibility qualities were evaluated using PSNR and it was found to be 40. The embedding using the average process requires modifying two low frequency locations to embed one bit of the watermark. Better methods can be used such as modifying the coefficients into odd or even instead of averaging.

Another blind image watermarking technique based on the DC component in DCT is proposed in [57]. The original image is divided into 8×8 sub-blocks and the DCT is performed for every block of the image. The watermark information is embedded in the DC component of each sub-block. Lena grey-scale image of size 512×512 was used as test image. The watermark was a 64×64 binary image and the average PSNR was found to be 44.2. However, using the DC components for embedding limits the watermark size. For example, the maximum watermark size that can be embedded when using a host of 512×512 is 64×64 .

A dual colour image DCT watermarking technique is proposed in [58]. The technique is blind. The original watermark image and the original host image in RGB colour model were converted to the YCbCr colour model. The block DCT was implemented in their Y,Cb and Cr layers respectively. After the 8×8 DCT transform and the zigzag sorting, the watermark sequences were embedded into the middle-low frequency coefficients of the host image. Lena colour image of size 512×512 was used as a host image. Two 64×64 colour watermarks of size 64×64 were used. The PSNR was used to evaluate the invisibility qualities of the method. The average PSNR was 41.1. Using another colour channel such as RGB could increase the invisibility qualities of the proposed method.

Another blind colour watermarking method is proposed in [59]. The watermark was embedded in the host image Cb's component. The watermark was first coded by Reed-Solomon code and embedded in the DCT middle frequency coefficients of the host image Cb component. Lena colour image of size 512×512 was used as host image. After embedding the watermark, the average PSNR was found to be 35.1. The extracted watermark in this method was not evaluated subjectively or objectively.

Another colour image watermarking method based on DCT is proposed in [60]. The original image was transformed to YCbCr colour model. The DCT was applied to

the Cb and Cr channels. After a process of zigzag, the middle frequency coefficients were modified to embed the watermark information. The technique was implemented to detect the watermark but not extract it. A possibility of false detection might occur.

In [24] a DCT technique to embed eight binary watermark bits in sixteen middle-frequency DCT coefficients of the 8×8 host image sub-blocks is introduced. This algorithm allows several watermarks to be embedded within the host image, which increases the robustness against cropping attacks and JPEG compression. However, using sixteen DCT coefficients extracted from the host image to hide eight watermark bits would further reduce the invisibility quality. This technique cannot survive any vertical cropping attack because of the spatial correlation between the host image sub-blocks and the sub-blocks of the watermark copies.

Another technique using DCT was proposed in [38]. The watermark was embedded in the DC components of the DCT coefficients of the 8×8 sub-blocks of the host image. Although this technique was shown to be robust to JPEG compression attacks, its performance with respect to other attacks is moderate since the watermark can only be embedded once in the host image. In addition, any significant cropping attacks (horizontal and/or vertical) would result in the loss of significant parts of the reconstructed watermark.

The authors in [39] presented a digital watermarking system based on vector quantization and DCT to embed a watermark in a grey-scale image. The algorithm can embed 16 times more information compared to other traditional DCT based watermarking systems. The recovered images have good visual quality. This system is robust only against the JPEG compression attack.

The DCT coefficients of the watermark image are embedded into the DCT domain of the host image by considering the similarity of DCT blocks' energy of both images in the algorithm proposed in [45]. The algorithm recovers the watermark without any reference to the original image with a proposed post proceeding error. An error correction mechanism is introduced to obtain higher quality watermarks. The experimental results showed that the proposed algorithm produces an average

PSNR value of 40. The robustness was evaluated against print-and-scan and various attacks. However, the technique was used only for grey-scale images.

The authors in [61, 62] proposed two watermarking techniques that are robust against JPEG compression attacks and were developed using Walsh sequences to encode the signatures before embedding them in the host images. The results showed that, regardless of the watermarking domain, the use of Walsh coding offered a significant improvement in robustness against JPEG compression compared to the case of no coding. Those techniques were non-blind and the robustness performances were measured against the JPEG compression attack only. These techniques were not evaluated against other attacks such as, filtering, cropping and additive noise.

Several watermarking algorithms [31, 63-66] were proposed to hide small amounts of useful data using small sized watermarks (e.g., 32×32). On the other hand, other researchers [67-71] proposed techniques using large watermarks.

2.3.3 Watermarking Techniques using mobile devices

The Polaroid Company announced its decision to cease the production of films for its instant cameras in 2009. This created a problem in that photographs taken with instant cameras are the only photo types that are unalterable and therefore useable as evidentiary photos. Since photos taken using digital cameras can be readily altered, they have minimal value for use as evidence in a court of law. Consequently, for a future world where instant cameras will become unusable, it is imperative to develop techniques that will make it possible to verify the authenticity of digital camera photos.

In the past few years, several watermarking schemes have been proposed, but digital watermarking schemes applied to mobile devices or digital cameras are scarce. In addition, watermark insertion is needed in devices with various features such as Digital cameras, PDAs and mobile phone as multimedia services on those devices are heavily used.

Some researchers [72-74] proposed watermarking algorithms that can be applied to mobile phones. A blind watermarking scheme for camera phone was proposed in [72] using wavelet decomposition. A set of grey-scale images which are obtained by

Samsung phone camera were used as host images. The watermark used consists of the phone number and the date. Based on the sizes of the watermark and the host image the robustness was assessed against only the JPEG compression attack. This attack was chosen because the output image captured by phone cameras is usually compressed by JPEG compression. This method is computationally complex and is not suitable for real-time applications since phone camera's are utilizing DCT for JPEG compression. The algorithm was tested by using only grey-scale images.

The paper in [73] proposes a technique for verifying the authenticity of photos taken with mobile phone cameras by embedding them with fragile digital watermarking. This technique simultaneously embeds the camera's ID number, the time the photo was taken, and the shooting location obtained by GPS as digital watermarking data. This makes it possible to verify not only a photograph's authenticity, but the conditions under which it was taken as well. The verification data was embedded in the DCT blocks obtained through the JPEG encoding process. The verification information (ID, Location and GPS) is expressed as binary numbers. Error detection coding was applied. The embedding process was applied by using odd and even difference between the maximum absolute value of the 64 coefficients in each DCT Block and a selected coefficient among the remaining 63. A secret key was used to determine the randomly selected coefficients. PSNR value was found to be 60 dB.

Another method for identifying the owner of mobile device who has taken the photographs or a movie through mobile phone is proposed method in [74]. Low-middle frequency coefficients of the DCT sub-blocks have been chosen for embedding. The method based on selecting two frequency coefficients and swapping them to embed the binary information in each DCT sub-block. The techniques were not evaluated against any attacks. Also, the visual quality of watermarked image was not evaluated or mentioned.

2.4 Final Remarks

- Most current watermarking techniques use pseudo-random sequences or binary text images as watermark data.
- Many watermarking algorithms focus mainly on watermarking of grey-scale images. However, the extension to colour images is sometimes

realised by utilising the blue component of the RGB colour model or Y channel of the YCrCb model.

- The size of the watermark is very small compared to the size of the host images. Large watermarks are rarely used.
- Multiple-embedding for a single watermark is uncommon in most of the watermarking algorithms.
- Some of the current watermarking algorithms use the middle AC components of the DCT.
- Stirmark was used to test the robustness of many of the used watermarking algorithms. Furthermore, most of them focus mainly on two or three attacks.
- Many watermarking methods are non-blind since they require the original image for extraction.
- Security and capacity are rarely evaluated or mentioned in the previous work.
- Some researchers proposed watermarking algorithms for greyscale images captured by phone cameras.
- The fidelity of the watermarked images were examined by using only the peak signal to noise ratio. Only few algorithms utilized the structural similarity index measurements for this purpose.

Based on the previous remarks, the direction of the proposed algorithms in this thesis will be implemented to rectify the shortcoming in some previous algorithms. One of the most important digital watermarking challenges is to design an embedding-extraction system that would not affect the quality of the image while at the same time, satisfy the conditions of security, capacity and robustness. Transparency, robustness, security and high capacity techniques will be the major direction of the proposed work in the next chapters (3 & 4) using grey-scale and colour images.

The efficient watermarking system must fulfill the common requirements of transparency and robustness. In addition, a new challenge arises which is a need for a fast and less complex embedding strategy. For this research, the aim is to develop an effective image watermarking algorithms that could satisfy most of the quality requirements and demand little computation to insert or extract. Thus, discrete cosine

transform will be chosen to reduce the computation complexity, since all the mobile phone cameras and digital cameras are already utilizing DCT for JPEG compression. This will simplify any future implementation.

Another challenge that will be tackled in this thesis is to make the watermarking system deal easily with different file format, different colour formats, different sizes and type of images, different sizes and types of watermarks and different applications.

Data capacity is yet another challenge for digital watermarking systems, in other words there is a need to determine how much data can be added before disturbing the quality of the image. The capability of any watermarking system to hide large amount of data as well as small data while maintaining the robustness and quality is yet an important challenge in watermarking.

Some digital cameras are small and convenient to use while still technologically superior in image quality. There have been claims within the media about the potential misuse of mobile phones with built in camera capability. This makes them easier to use in an unacceptable manner. Camera phones are designed to provide means of transferring images via mobile phone for business or personal reasons. It is indeed a difficult task to identify the user who has captured photographs or movies and made them public. Chapter 5 of this thesis focuses on a technique through which this problem can be solved.

Non-blind watermarking algorithms identify the watermark with the aid of the original image. Using non-blind watermarking algorithms will lead to immense number of original works being stored for extraction. Thus, blind techniques will be chosen in this research to facilitate future implementation of the proposed techniques in mobile devices or digital cameras.

A false positive is the identification of a watermark from a watermarked work which does not contain one in reality. Many current watermark schemes fail to resolve the rightful ownership as the embedded watermark is "detected". Multiple embedding and multiple "extraction" algorithms are chosen in this research to protect the ownership of digital images to overcome the false positive identification problem.

Chapter 3 Digital Watermarking Algorithms for Grey-scale Images

3.1 Overview

This chapter begins by introducing several host images and different kinds of binary watermarks. This section then is followed by investigation of several *complete (blind)* watermarking algorithms for still grey-scale images that work in the frequency domain using the discrete cosine transform (DCT). The proposed watermarking algorithms in this chapter are the starting point of the author's work related to this thesis. The aim is to prove the rightful ownership for the digital grey-scale images technically and integrate different watermarking requirements such as quality, robustness, security and capacity. The watermarking systems implemented in this thesis deal easily with different file formats, different colour formats (chapter 4), different sizes and type of images, different sizes and types of watermarks.

3.2 Host Images and Watermarks

Several grey-scale and colour test images were used throughout the course of this work to test the various algorithms. Some of the USC-SIPI image datasets [20, 21] such as Lena, Baboon, Peppers were used. Kodak images [22] are another dataset of standard images. Some of the Kodak images have also been used throughout this research. Other non-standard images captured by digital camera have been also used. A total of 40 colour images (*24 bits/pixel*) and their associated grey-scale images (*8 bits/pixel*) were used as shown in Figure 3.1. All images are of size 512×512 pixels. Also, handwritten signatures of different sizes were used as watermark images as shown in Figure 3.2.



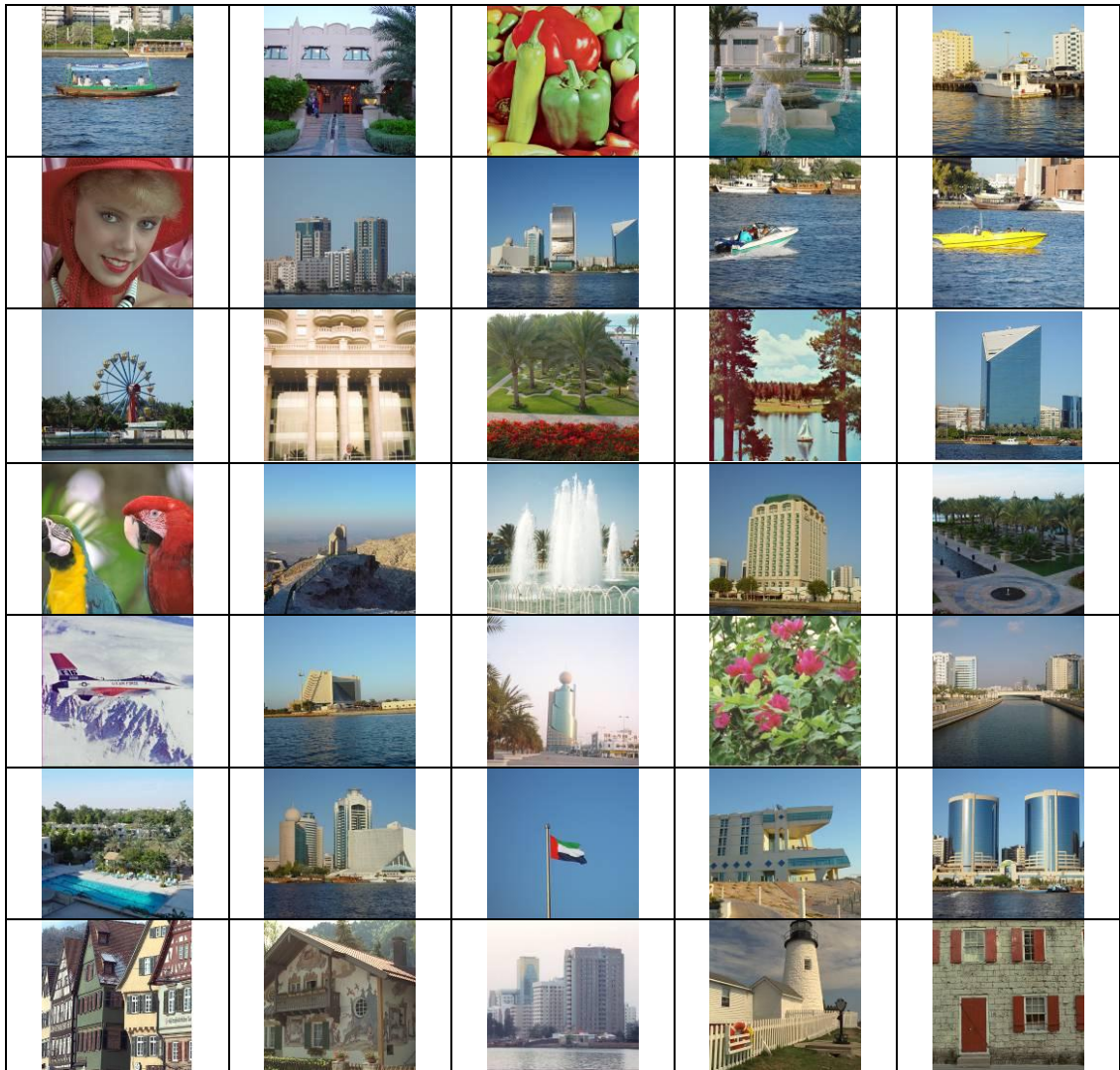


Figure 3-1 Host colour images which are also used in greyscale form.


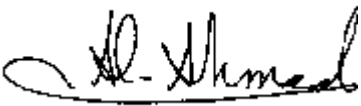

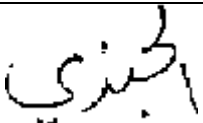



			
(a): Signature1 96x64	(b):Signature2 192x64	(c): Signature3 224x128	
			
(d): Signature4 96x64	(e): Signature5 224x96	(f): Signature6 64x64	(g): Signature7 32x32

Figure 3-2 Binary watermarks

3.3 Performance Evaluation and Testing Strategy

The efficient watermarking system must fulfill the common requirements of transparency and robustness. In order to monitor the response to the watermarking challenges described earlier in chapter 1, a testing strategy was applied to the algorithms proposed in this thesis. The performance evaluation and testing strategy consider the watermarking requirements, robustness, tradeoffs involved and speed.

- **Host Images and watermarks**

The proposed algorithms are tested using different images of size 512×512 . The proposed algorithms are also tested using high resolution images of size 1024×1024 and 2048×2048 . Different watermarks can be used in the proposed algorithms. Handwritten signatures of different sizes are mainly used as watermarks. Hand written signatures are used because they are visually recognizable patterns and are more intuitive for representing one's identity. Mobile phone numbers are used as watermarks in chapter 5. Note that the Lena image is used as a reference to evaluate the quality and the robustness in this thesis.

- **Evaluation metrics**

The transparency of the proposed techniques is evaluated. Two evaluation techniques are used in the experiments with different watermarking strengths Δ and different signatures size. The first evaluation is carried out by calculating the peak signal to noise ratio (PSNR) between the host image and the watermarked image. The second evaluation is carried out using the structural similarity index measurement (SSIM) between the host image and the watermarked image [25] (see section 2.3 for details). The higher the SSIM percentage is, the larger the similarity between the compared images. To verify the robustness of the proposed method, various common signal processing and geometric attacks are applied to the watermarked images. The normalized correlation (NC) is used to measure the similarity between the original and the extracted watermarks.

- **Watermark embedding strength**

The goal of the proposed algorithms are to embed digital watermarks that are both imperceptible to the human eye and robust against attacks. This can be a delicate balancing act, since the robustness and visibility of a digital watermark are directly

related. An increase in the watermark embedding strength also increases the visibility of the watermark. As a part of the testing strategy, various embedding strengths have been investigated to determine which values provide best performance for the majority of the images.

- **Capacity**

The capability of any watermarking system to hide large or small amounts of data while maintaining the robustness and quality is an important challenge in watermarking. The smaller the number of pixels in the watermark image, the more the watermark can be repeated throughout the host image which in turn increases the robustness. The performance evaluation will also consider and show by the experimental testing the effect of watermark size in the proposed techniques.

- **Speed**

Speed is very dependent on the type of implementation: software or hardware. Software implementation depends on the hardware used to run it. As a part of the evaluation scheme, the time measurement has been undertaken under different watermark sizes.

3.4 Algorithm 1: A Blind Image Watermarking of Handwritten Signatures Using Low-Frequency Band DCT Coefficients

The proposed embedding algorithm here is totally blind as the original host image is not required for the watermark extraction. The watermark data is embedded in the very low-frequency components of the DCT-domain. This range of frequencies is chosen because the high frequency components may be discarded in some image processing operation such as JPEG compression. Placing the watermark in the very low DCT coefficients maximizes the chances of reconstructing the watermark even after common signal distortions. Furthermore, modification of these components results in severe image degradation long before the watermark itself is destroyed. An attacker would have to add much more noise energy in order to sufficiently remove the watermark. However, this process would destroy the image fidelity. A shuffle scheme is applied for each binary watermark copy before embedding by representing

the watermark in a vector format and applying a shift operation to this vector. The shuffle scheme is necessary to reduce the spatial correlation between the watermark and the host image. Hand written signatures were used as watermarks since they are visually recognizable patterns and more intuitive for representing one's identity. The watermark is further protected by using a secret key. Multiple copies of the same signature were embedded in the host image. This will increase the robustness of the watermark against several attacks since each watermark will be individually reconstructed and verified before applying an averaging process.

3.4.1 Proposed Robust Image Watermark Algorithm

The proposed watermarking scheme is based on the possibility of embedding multiple copies of the same binary watermark(s) in the host image. This will increase the robustness of the watermark against several attacks.

Assume that $f(i, j)$ is the host image of size Z_h pixels and $w(i, j)$ is the binary watermark of size Z_w bits which is usually much smaller compared to the size of the host image. Also assume that the size of the host image can accommodate multiple copies of the watermark image. The watermark is converted into a vector of size $1 \times Z_w$ and the host image is divided into N_{HB} non-overlapping 8×8 sub-blocks. The numbers of watermark copies n that can be embedded in the host image can be given by:

$$n = N_{HB} / N_{wB} \quad (3.1)$$

Where N_{wB} is the numbers of the watermark sub-blocks, which can be defined as:

$$N_{wB} = Z_w / B_n \quad (3.2)$$

Where Z_w is the watermark size and B_n is the number of watermarks bits that can be embedded in each N_{HB} (for example, 8 bits per block).

3.4.2 The Embedding Process

The proposed embedding algorithm as shown in Figure 3.3 can be described as follows:

Step 1: The host image is divided into 8×8 sub-blocks and is DCT transformed. So

$$F_k(u, v) = DCT\{f_k(i, j)\}, \quad 1 \leq k \leq N_{HB} \quad (3.3)$$

Inside each 8×8 sub-block, eight DCT coefficients are identified as shown in Figure 3.4. These eight coefficients in each 8×8 sub block are denoted by

$$U_c, 1 \leq c \leq 8 \quad (3.4)$$

Step 2: The binary watermark is converted into a vector of size $1 \times Z_w$ and then it is randomly scrambled using a secret key. This scrambling process is essential to reduce the spatial correlation between the host image and the embedded watermark.

Step 3: The watermark is divided into $N_{wB} 1 \times 8$ sub-blocks. Each watermark sub-block would be embedded into one of the 8×8 sub-blocks of the host image. So one complete copy of the watermark would require $N_{wB} 8 \times 8$ sub-blocks. The bit embedding equation can be defined as follows:

If $w(i,j)=1$ then

$$F_k(u, v) = \begin{cases} \Delta Q_e \left(\frac{F_k(u, v)}{\Delta} \right) & u, v \in U_c \quad 1 \leq k \leq N_{HB} \\ F_k(u, v) & u, v \notin U_c \quad 1 \leq k \leq N_{HB} \end{cases} \quad (3.5)$$

If $w(i,j)=0$ then

$$F_k(u, v) = \begin{cases} \Delta Q_o \left(\frac{F_k(u, v)}{\Delta} \right) & u, v \in U_c \quad 1 \leq k \leq N_{HB} \\ F_k(u, v) & u, v \notin U_c \quad 1 \leq k \leq N_{HB} \end{cases}$$

Where Δ is a scaling quantity and it is also the quantization step used to quantize either to the even or odd number, Q_e is the quantization to the nearest even number and Q_o is the quantization to the nearest odd number.

Step: 4: Step 3 is repeated n times to embed the n copies of the watermark in all N_{HB} of the host image. But before repeating step 3, a shuffle scheme is applied for each watermark copy before the embedding process. The shuffling scheme will be discussed in next section.

Step 5: The watermarked host image is obtained by using the inverse DCT transform.

$$f_k(i, j) = IDCT\{F_k(u, v)\}, \quad 1 \leq k \leq N_{HB} \quad (3.6)$$

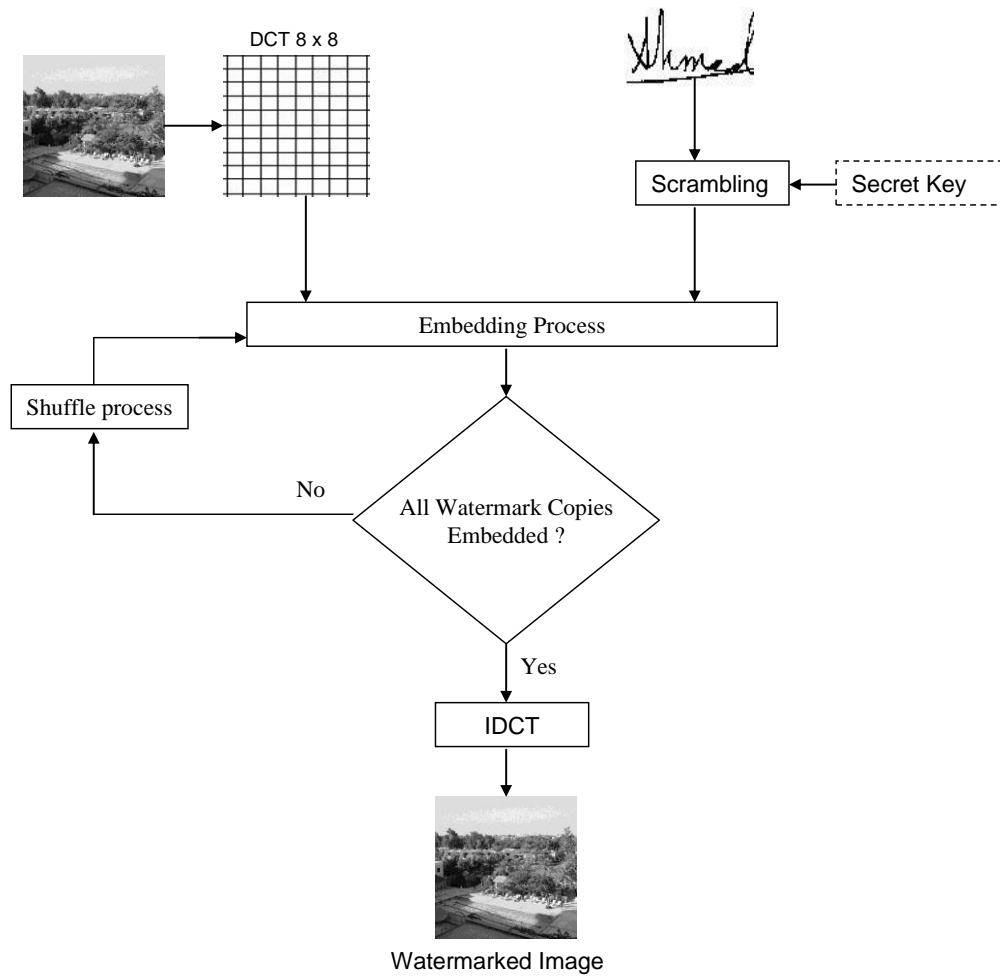


Figure 3-3 A flow graph for the embedding process.

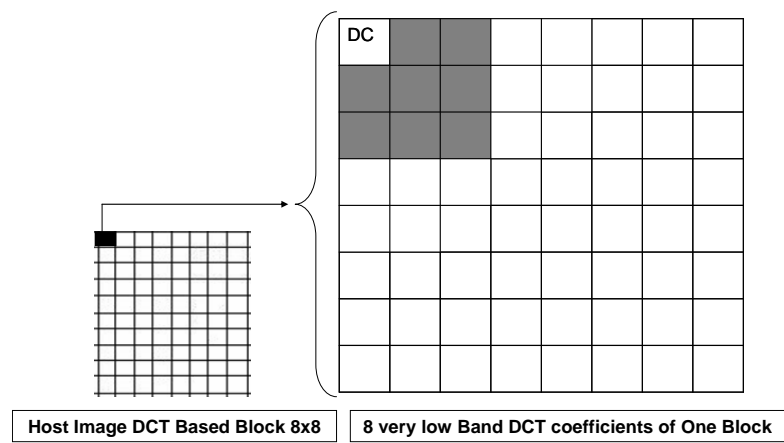


Figure 3-4 Used DCT coefficients in one 8x8 block of the host image.

3.4.3 The shuffle Scheme

The shuffle scheme is important to shield the proposed algorithms against cropping attacks. The shuffle scheme is applied to each watermark copy before embedding. A flow-graph of the proposed algorithm is shown in Figure 3.5. Since all the proposed watermarking schemes in this research are based on the possibility of embedding multiple copies of the same binary watermark in the host image, then each watermark copy is differently shuffled before the embedding process. This can be done by representing the watermark copy as a vector and applying multiple-shift operations before the embedding process. The shift operation must be circular. This shift is necessary to reduce the spatial relation and to increase the robustness against vertical cropping attacks. The number of watermark shifted bits depends on the host image size and the watermark size. It can be calculated as follows:

$$w_{SB} = Z_w / n \quad (3.7)$$

Where w_{SB} is the number of watermark shifted bits, Z_w is the size of the watermarked image and finally n is the number of watermark copies to be embedded in the host image.

The watermark information can be retrieved by using a reverse process to the shuffle scheme. This will yield the original bits order. The proposed shuffle scheme may be applied also for other techniques that allow the possibility of embedding multiple copies of the watermark in the host image. It is worth mentioning that although the proposed scheme is blind and does not require the original host image for reconstruction, it still requires information such as the sizes of both the host and watermark images and the watermarking strength Δ . The shuffle scheme has been tested in the presented algorithm and the results are shown in Table 3.1.

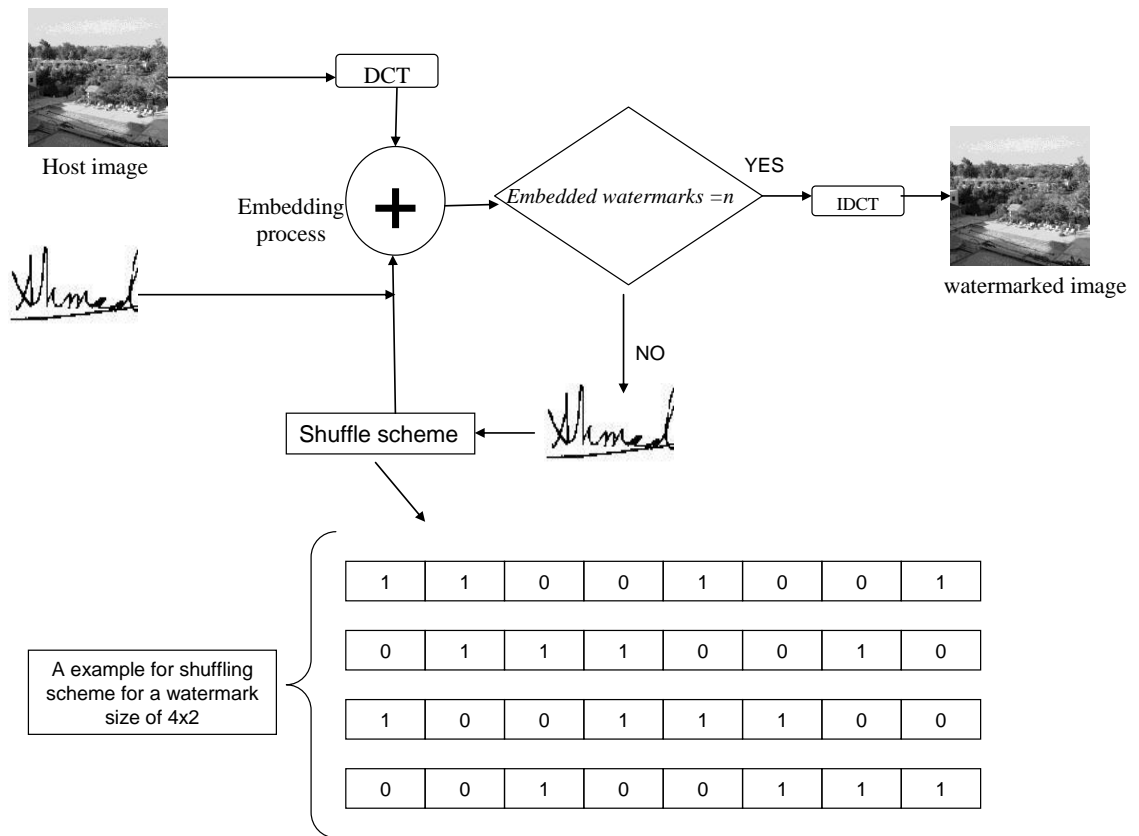


Figure 3-5 A flow-graph of the proposed shuffle scheme

Table 3-1 Effect of using shuffle scheme against cropping attack

(a): 50% cropping	(b): 75% cropping	(c): 75% V&H cropping	(d): 50% cropping	(e): 75% cropping
Results of cropping attacks using shuffle scheme				
MSE=0.0021	MSE=0.0563	MSE=0.0760	MSE=0.0050	MSE=0.0098
Results of cropping attacks without using shuffle scheme				
MSE=0.0537	MSE=0.0828	MSE=0.0850	MSE=0.0052	MSE=0.0099

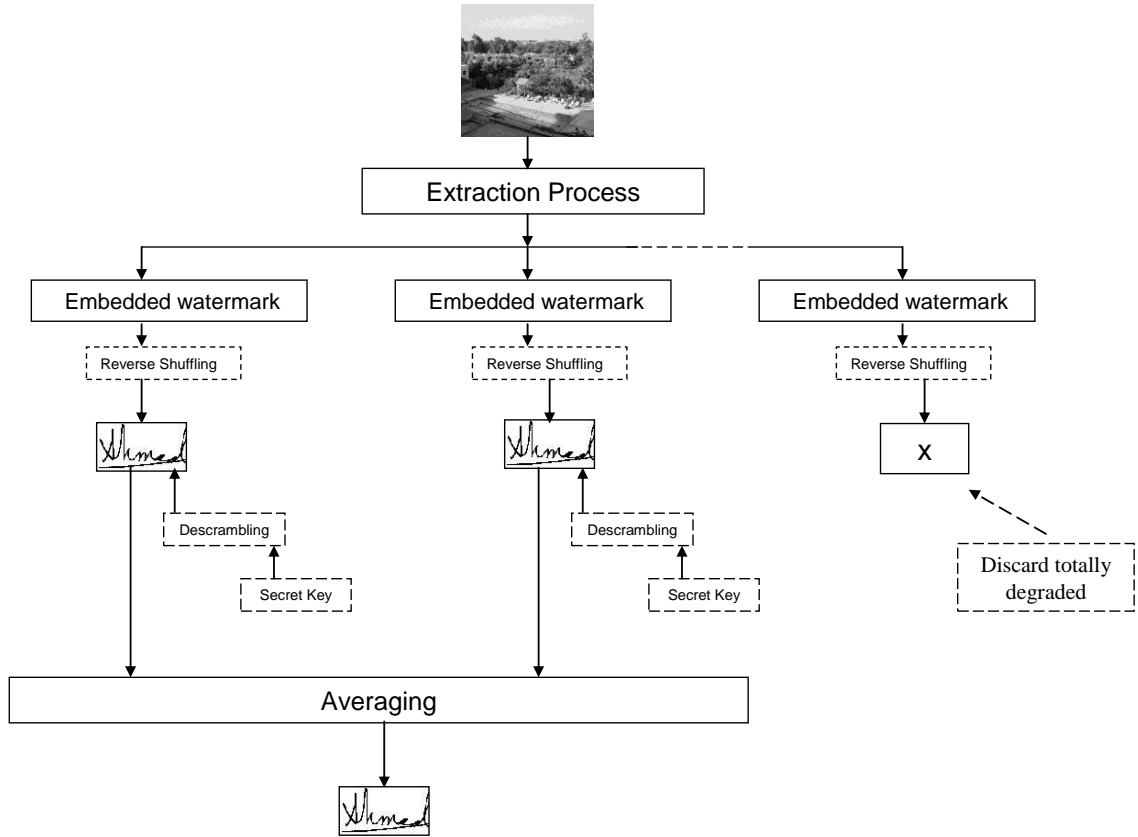


Figure 3-6 A block diagram of the extraction process

3.4.4 The Reconstruction Process

The embedded watermarks information $w(i, j)$ can be extracted by performing 8×8 DCT transform for the watermarked host image and then indicating the same coefficients of the host image that carries the 8 bits of the embedded watermarks. A reverse shuffling scheme is implemented for the reconstructed watermarks. By using the same secret key in the initial scrambling operation, the scrambled watermarks are descrambled to get the original watermarks. The extraction process is performed without needing the original unmarked image. Simply the recovery process is the inverse of the embedding process. Each predefined frequency coefficient is quantized by Δ and rounded to the nearest integer. The extracted formula is defined as follows:

$$\begin{aligned}
 & \text{If } Q\left(\frac{F_k(u, v)}{\Delta}\right) \text{ odd then } w(i, j) = 0 \\
 & \text{If } Q\left(\frac{F_k(u, v)}{\Delta}\right) \text{ even then } w(i, j) = 1
 \end{aligned} \tag{3.8}$$

Where Q is rounded to the nearest integer. The value of Δ is the same as the one used in the embedding process. A visual representation for the extraction process is shown in Figure 3.6. Afterwards, totally degraded copies of the extracted watermarks are discarded. The totally degraded copies are identified when a totally black watermark is extracted after the horizontal cropping operation. The average is calculated by summing the resultant watermark copies divided by their number. The user can choose one copy of the watermark as the final watermark if it provides better results than the resultant average watermark. The following information must be present for the extraction process: the size of the host image, the size of the watermark image and the watermark embedding strength Δ . Figure 3.7 illustrates the averaging process when the watermarked images has been cropped horizontally by 50%. The first part of the image was not attacked by cropping and two watermarks have been fully reconstructed, the third watermark has been reconstructed with errors, finally the last two watermarks have been totally degraded due to the cropped area of the watermarked host image. Each watermark has been individually reconstructed and tested before the averaging process. Figure 3.8 shows the extraction process of the embedded watermarks after 50% JPEG compression. The reconstructed watermarks were tested in order to ensure that it is not fully degraded. The averaging process has managed to reduce the error as demonstrated by the normalized correlation (NC) values.

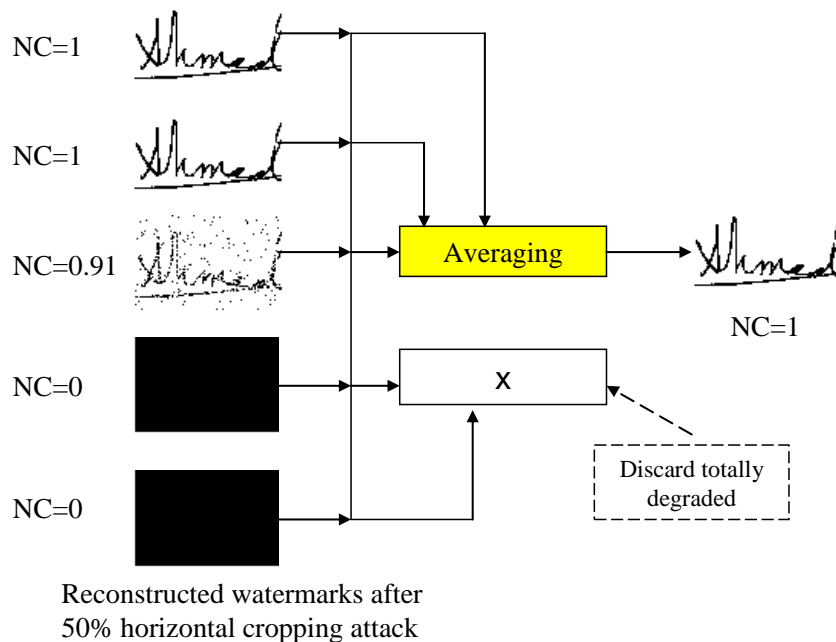


Figure 3-7 The averaging and correction process after 50% horizontal cropping

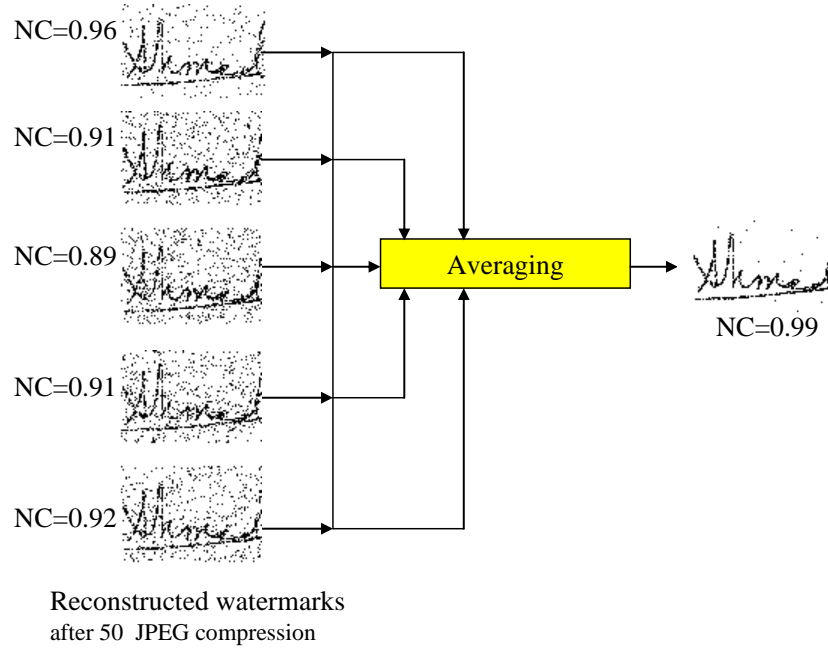


Figure 3-8 The averaging and correction process after JPEG 50 compression attack

3.4.5 Results

This algorithm is examined using different images of size 512×512 . The same handwritten signatures of 3 different sizes 96×64 , 64×64 and 32×32 are used as watermarks Table 3.2 demonstrates the perceptual invisibility of the proposed algorithm at different embedding strengths and different watermarks size. The average PSNR values between the watermarked and original images using a watermark size of 32×32 are 44.2902 dB and 38.7227 dB for watermarking strengths $\Delta = 8$ and $\Delta = 16$, respectively. If watermarks of size 64×64 are used, then the average PSNR values will be 44.4558 dB and 39.2518 dB for $\Delta = 8$ and $\Delta = 16$, respectively. Finally, the average PSNR values using a watermark size of 96×64 are 44.4719 dB and 39.2852 dB for $\Delta = 8$ and $\Delta = 16$, respectively. Hence, it can be concluded that the technique will not degrade the quality of the original host image when larger watermarks are used.

In Table 3.3, the perceptual invisibility of the proposed algorithm is evaluated using SSIM at different embedding strengths and different watermarks size. The original “Lena” colour image has been used to examine the perceptual quality at different embedding strengths as depicted in Table 3.4.

Visual inspection of the extracted watermarks using different embedding strengths at $\Delta = 14$ and $\Delta = 16$ are depicted in Tables 3.7 and 3.8 respectively. The experimental results show that the performance achieved by the proposed method for the extracted watermark after running through attacks is perceptually visible at $\Delta = 14$, which can be considered as the best value for embedding strength, Higher embedding strength value such as $\Delta = 16$ will provide strong robustness and distinct perceptual visibility for the extracted watermark. It is worth noting that higher embedding strength could reduce the invisibility qualities as demonstrated in Tables 3.2 and 3.3. The smaller the number of pixels in the watermark image, the more the watermark can be repeated throughout the host image which in turn increases the robustness. Table 3.5 shows the effect of watermarks size in the proposed technique.

The execution time of the Matlab algorithms on a 2 Ghz Centrino processor and 1 Ghz memory is shown in Table 3.9. It should be noted that the real processing time will depend on the processor used. Finally, the performance evaluation against high resolution images is illustrated in Table 3.10.

Table 3-2 PSNR for different grey-scale images with different embedding strengths and watermark sizes

Watermark size 96 × 64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
PSNR at $\Delta = 8$	44.4719	45.2637	43.8883
PSNR at $\Delta = 12$	41.4030	42.2542	40.4077
PSNR at $\Delta = 14$	40.2738	40.9971	39.0659
PSNR at $\Delta = 16$	39.2852	40.0303	37.9413
Watermark size 64 × 64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
PSNR at $\Delta = 8$	44.4558	45.2162	43.8759
PSNR at $\Delta = 12$	41.3635	42.1179	40.4144
PSNR at $\Delta = 14$	40.2426	40.9275	39.0568
PSNR at $\Delta = 16$	39.2518	39.9402	37.9484
Watermark size 32 × 32			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
PSNR at $\Delta = 8$	44.2902	44.6718	43.9108
PSNR at $\Delta = 12$	41.0031	41.2865	40.3735
PSNR at $\Delta = 14$	39.7804	39.9930	39.0397
PSNR at $\Delta = 16$	38.7227	38.9344	37.9323

Table 3-3 SSIM for different grey-scale images with different embedding strengths and watermark sizes

Watermark size 96 × 64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
SSIM at $\Delta = 8$	0.9859	0.9847	0.9943
SSIM at $\Delta = 12$	0.9739	0.9709	0.9877
SSIM at $\Delta = 14$	0.9671	0.9620	0.9836
SSIM at $\Delta = 16$	0.9552	0.9529	0.9794
Watermark size 64 × 64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
SSIM at $\Delta = 8$	0.9858	0.9845	0.9942
SSIM at $\Delta = 12$	0.9733	0.9696	0.9877
SSIM at $\Delta = 14$	0.9666	0.9608	0.9834
SSIM at $\Delta = 16$	0.9585	0.9524	0.9793
Watermark size 32 × 32			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
SSIM at $\Delta = 8$	0.9848	0.9812	0.9943
SSIM at $\Delta = 12$	0.9692	0.9608	0.9877
SSIM at $\Delta = 14$	0.9602	0.9478	0.9832
SSIM at $\Delta = 16$	0.9491	0.9351	0.9791

Table 3-4 Original and watermarked Lena images at different embedding strengths
















Table 3-5 Normalized correlation for Lena grey-scale image

Watermark size 96×64 , at $\Delta = 14$.			
Attacks	NC	Attacks	NC
Cropping 75% V	0.9696	Low pass 3×3	0.9768
Cropping 50% V	0.9709	Wiener 3×3	0.9943
Cropping 75% H	1	Median 3×3	0.9887
Cropping 75% V+ H	0.9858	JPEG 75	1
Scale 2	1	JPEG 50	1
Scale 0.75	0.9731	JPEG 30	0.9578
Scale 0.5	0.8876	Gaussian noise $m=0, v=0.002$	0.6303
S&P noise, $d=0.02$	0.7299	Gaussian noise $m=0, v=0.001$	0.7765
Contrast enhancements intensity=0.3, 0.9	0.9885	S&P noise, $d=0.02$ + Median 3×3	0.9911
Contrast enhancements intensity=0.1, 0.5	0.9064	S&P noise, $d=0.05$ + Median 3×3	0.9816
Stirmark_AFFINE_1	0.8985	Stirmark_CONV_1	0.9500
Stirmark_AFFINE_8	0.8199	Stirmark_RML_10	0.9615
Stirmark_ROTSCALE_0.25	0.9057	Stirmark_RML_50	0.9903
Stirmark_ROTSCALE_-0.5	0.8204	Stirmark_RML_100	0.9951
Stirmark_ROT_-0.5	0.8155	Stirmark_ROT_0.25	0.9095
Stirmark_ROTROP_-0.5	0.8170	Stirmark_ROTROP_0.25	0.9188
Watermark size 64×64 , at $\Delta = 14$.			
Attacks	NC	Attacks	NC
Cropping 75% V	0.9770	Low pass 3×3	0.9853
Cropping 50% V	0.9801	Wiener 3×3	0.9992
Cropping 75% H	1	Median 3×3	0.9961
Cropping 75% V+ H	0.9888	JPEG 75	1
Scale 2	1	JPEG 50	1
Scale 0.75	0.9875	JPEG 30	0.9837
Scale 0.5	0.9438	Gaussian noise $m=0, v=0.002$	0.6886
S&P noise, $d=0.02$	0.8764	Gaussian noise $m=0, v=0.001$	0.8567
Contrast enhancements intensity=0.3, 0.9	0.9897	S&P noise, $d=0.02$ + Median 3×3	0.9925
Contrast enhancements intensity=0.1, 0.5	0.9446	S&P noise, $d=0.05$ + Median 3×3	0.9914
Stirmark_AFFINE_1	0.9575	Stirmark_CONV_1	0.9699
Stirmark_AFFINE_8	0.9070	Stirmark_RML_10	0.9754
Stirmark_ROTSCALE_0.25	0.9547	Stirmark_RML_50	0.9793
Stirmark_ROTSCALE_-0.5	0.9062	Stirmark_RML_100	0.9879
Stirmark_ROT_-0.5	0.9037	Stirmark_ROT_0.25	0.9680
Stirmark_ROTROP_-0.5	0.9051	Stirmark_ROTROP_0.25	0.9650

Watermark size 32×32, at $\Delta = 14$.			
Attacks	NC	Attacks	NC
Cropping 75% V	0.9901	Low pass 3×3	1
Cropping 50% V	0.9961	Wiener 3×3	1
Cropping 75% H	1	Median 3×3	1
Cropping 75% V+ H	0.9921	JPEG 75	1
Scale 2	1	JPEG 50	1
Scale 0.75	1	JPEG 35	1
Scale 0.5	0.9832	Gaussian noise m=0, v=0.002	0.6960
S&P noise, d=0.02	0.9586	Gaussian noise m=0, v=0.001	0.9586
Contrast enhancements intensity=0.3, 0.9	1	S&P noise, d=0.02+ Median 3×3	1
Contrast enhancements intensity=0.1, 0.5	0.9845	S&P noise, d=0.05+ Median 3×3	1
Stirmark_AFFINE_1	0.9913	Stirmark_CONV_1	1
Stirmark_AFFINE_8	0.9770	Stirmark_RML_10	1
Stirmark_ROTSCALE_0.25	0.9989	Stirmark_RML_50	1
Stirmark_ROTSCALE_-0.5	0.9683	Stirmark_RML_100	1
Stirmark_ROT_-0.5	0.9607	Stirmark_ROT_0.25	0.9978
Stirmark_ROTROP_-0.5	0.9727	Stirmark_ROTROP_0.25	0.9967

Table 3-6 Watermarked images after attacks

		
Cropping 75% V	Low pass 3×3	JPEG 75
		
Cropping 50% V	Wiener 3×3	JPEG 50
		
Cropping 75% H	Median 3×3	JPEG 30
		
Cropping 75% V+ H	S&P noise, d=0.02	Gaussian noise m=0, v=0.002
		
Scale 2	Scale 0.5	Gaussian noise m=0, v=0.001
		
S&P noise, d=0.05+ Median 3×3	Contrast enhancements intensity=0.3, 0.9	Contrast enhancements intensity=0.1, 0.5

Stirmark_AFFINE_1	Stirmark_AFFINE_8	Stirmark_CONV_1
Stirmark_ROTSCALE_-0.5	Stirmark_ROTSCALE_0.25	Stirmark_ROT_0.25
Stirmark_ROT_-0.5	Stirmark_ROTSCROP_-0.5	Stirmark_ROTSCROP_0.25
Stirmark_RML_10	Stirmark_RML_50	Stirmark_RML_100

Table 3-7 Extracted watermarks after attacks at watermark embedding strength $\Delta = 14$

Cropping 75% V	Low pass 3x3	JPEG 75
Cropping 50% V	Wiener 3x3	JPEG 50
Cropping 75% H	Median 3x3	JPEG 30
Cropping 75% V+H	S&P noise, $d=0.02$	Gaussian noise $m=0, v=0.002$
Scale 2	Scale 0.5	Gaussian noise $m=0, v=0.001$

S&P noise, $d=0.05+$ Median 3x3	Contrast enhancements intensity=0.3, 0.9	Contrast enhancements intensity=0.1, 0.5
Stirmark_AFFINE_1	Stirmark_AFFINE_8	Stirmark_CONV_1
Stirmark_ROTSCALE_-0.5	Stirmark_ROTSCALE_0.25	Stirmark_ROT_0.25
Stirmark_ROT_-0.5	Stirmark_ROTROP_-0.5	Stirmark_ROTROP_0.25
Stirmark_RML_10	Stirmark_RML_50	Stirmark_RML_100

Table 3-8 Extracted watermarks after attacks at watermark embedding strength $\Delta = 16$

Cropping 75% V	Low pass 3x3	JPEG 75
Cropping 50% V	Wiener 3x3	JPEG 50
Cropping 75% H	Median 3x3	JPEG 30
Cropping 75% V+H	S&P noise, $d=0.02$	Gaussian noise $m=0$, $v=0.002$
Scale 2	Scale 0.5	Gaussian noise $m=0$, $v=0.001$


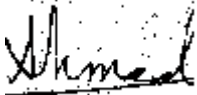





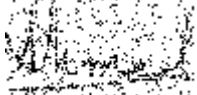





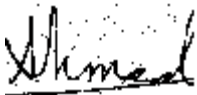




		
S&P noise, $d=0.05+$ Median 3×3	Contrast enhancements intensity=0.3, 0.9	Contrast enhancements intensity=0.1, 0.5
		
Stirmark_AFFINE_1	Stirmark_AFFINE_8	Stirmark_CONV_1
		
Stirmark_ROTSCALE_-0.5	Stirmark_ROTSCALE_0.25	Stirmark_ROT_0.25
		
Stirmark_ROT_-0.5	Stirmark_ROTROP_-0.5	Stirmark_ROTROP_0.25
		
Stirmark_RML_10	Stirmark_RML_50	Stirmark_RML_100

Table 3-9 Matlab execution time

Intel processor centrino 2 GHZ, RAM= 1 GB			
<i>Watermark size</i>	<i>Embedding time</i>	<i>Extraction time</i>	<i>Total time</i>
96×64	2.45 seconds	1.56 seconds	4.01 seconds
64×64	2.42 seconds	1.40 seconds	3.82 seconds
32×32	2.40 seconds	2.50 seconds	4.90 seconds

Table 3-10 Performance evaluation against high resolution images

Watermark size 96×64			
		(a): Original Host image	
			
(b): Watermarked host image of size 1024×1024		(c): Watermarked host image of size 2048×2048	
PSNR	SSIM	PSNR	SSIM
34.5	0.9745	29.01	0.9712
Attacks	host image size 1024×1024	host image size 2048×2048	
JPEG 30	NC= 0.9995	NC= 1.0000	
Cropping 75 % both sides	NC=0.9723	NC=0.9701	
Low-pass Filter 3×3	NC=0.9997	NC=1.0000	
Median filter 3×3	NC=0.9996	NC=1.0000	
Wiener filter 3×3	NC=1.0000	NC=1.0000	

3.5 Algorithm 2: An Adaptive Secured Watermarking Algorithm

A pirate may compare several watermarked images to detect the embedding locations and attack or remove a valid watermark. The location of the watermark in algorithm 1 was fixed (i.e. the first 8 AC coefficients) which compromised the security of the technique. In algorithm 2 the first 16 low frequency coefficients (excluding the DC value) in the 8×8 DCT block are screened and the eight coefficients with the maximum magnitudes are selected for embedding. The selected coefficients in algorithm 2 are adaptive and related to host image information. This was the motivation for developing algorithm 2 while maintaining the robustness and the invisibility qualities of algorithm 1.

This range of frequencies is chosen because the high frequency components may be discarded in some image processing operation such as JPEG compression. Placing the watermark in the low DCT coefficients maximizes the chances of reconstructing the watermark even after common signal distortions. Furthermore, any modifications to these components will result in severe image degradation, long before the watermark itself is destroyed. An attacker would have to add very large noise energy in order to sufficiently remove the watermark and this process would destroy the image fidelity.

The shuffle scheme is applied for each binary watermark copy before embedding by representing the watermark in a vector format and applying a shift operation to this vector. The shuffle scheme is necessary to reduce the spatial correlation between the watermark and the image. This has managed to increase the robustness against vertical cropping attacks. Hand written signatures were used again for the reasons mentioned in the previous section. The watermark is further protected by using a secret key. The algorithm used is blind. Multiple copies of the signature were embedded in the host image. This will increase the robustness of the watermark against several attacks since each watermark will be individually reconstructed and verified before applying an averaging process.

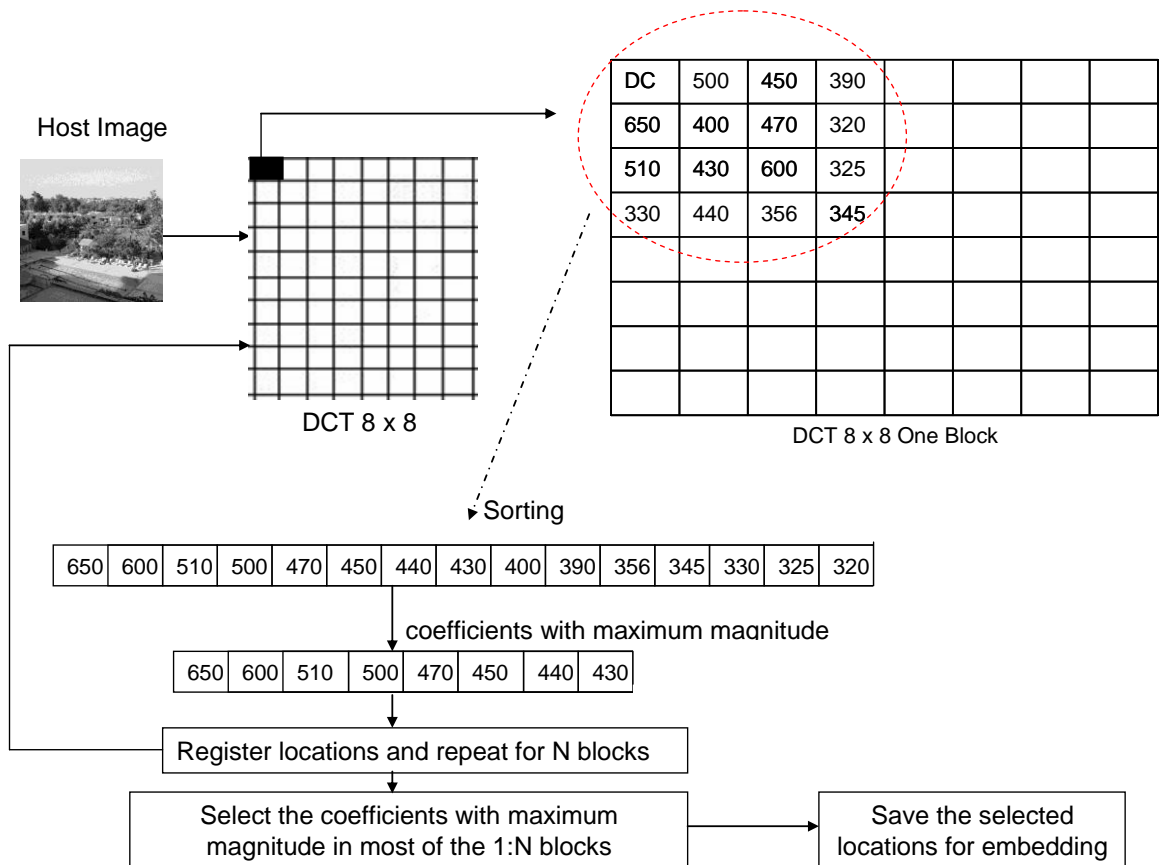


Figure 3-9 A flow graph representing the DCS process

3.5.1 The DCT Coefficients Selection (DCS) Process










An adaptive DCS process is applied to the host image. The DCS process managed to increase the security and reduce the visual changes after embedding the watermark. This process observes the perceptual capacity of the low frequency coefficients inside each of the DCT blocks to select the best 8 coefficients. After scanning the best resultant coefficients from each block, 8 coefficients will be used representing the majority of the best coefficients for all the DCT blocks, as will be explained in the next paragraph. The location of the 8 selected coefficients will be an added security to this scheme.

The DCT block consists of 8x8 coefficients. The 16 lower frequencies excluding DC are screened to find the coefficients with the highest magnitude and register their locations. This process is repeated for all the DCT blocks. The locations which are repeated more are selected. These locations will vary from one image to another according to the spatial frequency contents of the image. Eight binary bits of the

watermark will be embedded in these locations. A flow graph of the DCS process is shown in Figures 3.9

In order to test the security of the DCS process the images were screened again after embedding to verify that the method is secure and an attacker would not be able to use the DCS process again to detect the originally selected locations. Scanning the DCT blocks again after embedding would result in totally different locations from the previously registered locations in the original un-watermarked images as shown in Tables 3.10 and 3.11.

Table 3-11 The original DCS locations before embedding the watermark

The original DCS locations for some images before embedding the watermark									
<i>Image</i>									
	(1,2)	(2,1)	(2,2)	(1,3)	(1,4)	(2,3)	(3,1)	(3,2)	
	(1,2)	(2,1)	(2,2)	(1,3)	(3,1)	(1,4)	(2,3)	(3,2)	
	(1,2)	(2,1)	(2,2)	(1,3)	(3,1)	(2,3)	(3,2)	(1,4)	
	(2,1)	(1,2)	(3,1)	(4,1)	(2,2)	(1,3)	(3,2)	(4,2)	
	(1,2)	(2,1)	(2,2)	(3,1)	(1,3)	(3,2)	(2,3)	(4,1)	
	(2,1)	(1,2)	(1,3)	(2,2)	(3,1)	(1,4)	(4,1)	(2,3)	
	(2,1)	(1,2)	(2,2)	(1,3)	(3,1)	(1,4)	(2,3)	(3,2)	
	(2,1)	(1,2)	(3,1)	(2,2)	(4,1)	(1,3)	(3,2)	(2,3)	
	(2,1)	(1,2)	(3,1)	(4,1)	(1,3)	(3,2)	(2,2)	(1,4)	















		(2,1)	(1,2)	(3,1)	(4,1)	(1,3)	(2,2)	(3,2)	(1,4)
		(2,1)	(1,2)	(2,2)	(3,1)	(3,2)	(4,1)	(1,3)	(2,3)
		(1,2)	(2,1)	(2,2)	(1,3)	(3,1)	(3,2)	(2,3)	(1,4)

Table 3-12 The new DCS locations after embedding the watermarks

The new DCS locations for some images after embedding the watermark									
Image									
		(3,3)	(2,4)	(3,4)	(1,2)	(4,1)	(4,2)	(4,3)	(4,4)
		(4,1)	(3,3)	(4,2)	(2,4)	(3,4)	(4,3)	(1,2)	(4,4)
		(3,3)	(4,1)	(2,4)	(4,2)	(3,4)	(4,3)	(4,4)	(1,2)
		(2,3)	(3,3)	(1,4)	(4,3)	(2,4)	(3,4)	(4,4)	(2,1)
		(1,2)	(2,1)	(2,2)	(3,1)	(4,2)	(3,3)	(1,3)	(3,2)
		(1,2)	(2,1)	(1,3)	(3,2)	(2,4)	(4,2)	(3,1)	(3,3)
		(3,3)	(4,1)	(2,4)	(3,4)	(4,2)	(4,3)	(4,4)	(1,2)
		(3,3)	(1,4)	(2,4)	(2,1)	(4,2)	(3,4)	(4,3)	(1,2)
		(4,2)	(2,3)	(1,2)	(3,3)	(3,1)	(2,4)	(4,3)	(2,1)
		(4,2)	(2,3)	(3,3)	(2,4)	(3,4)	(4,3)	(4,4)	(2,1)

		(4,2)	(3,3)	(2,1)	(1,4)	(2,4)	(3,4)	(4,3)	(3,1)
		(1,2)	(2,4)	(3,3)	(2,1)	(3,4)	(4,1)	(4,2)	(4,4)

3.5.2 Results

This algorithm is examined using different images of size 512×512 . Also handwritten signature of size 96×64 is used as a watermark. Table 3.13 demonstrates the perceptual invisibility of the proposed algorithm at different embedding strengths and different watermarks' sizes. In Table 3.14, the perceptual invisibility of the proposed algorithm is evaluated using SSIM at different embedding strengths. The original "Lena" colour image was used to examine the perceptual quality at different embedding strengths as shown in Table 3.15. To verify the robustness of the proposed method, various common signal processing and geometric attacks are applied to the watermarked images. NC was used to measure the similarity between the original and the extracted watermarks as shown in Tables 3.16.

An increase in watermark embedding strength also increases the visibility of the watermark. Visual inspection of the extracted watermarks using embedding strengths at $\Delta = 14$ and $\Delta = 16$ are depicted in Tables 3.18 and 3.19.

The execution time of the Matlab algorithms on a 2 GHz Centrino processor and 1 Mb memory is shown in Table 3.12. Finally, the performance evaluation against high resolution images is illustrated in Table 3.21

Table 3-13 PSNR for different grey-scale images with different embedding strengths and watermark sizes

Watermark size 96 × 64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
PSNR at $\Delta = 8$	44.2569	45.2087	43.7658
PSNR at $\Delta = 12$	41.4030	42.0529	40.3897
PSNR at $\Delta = 14$	40.2738	40.7893	39.0089
PSNR at $\Delta = 16$	39.2852	40.0001	37.8765
Watermark size 64 × 64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
PSNR at $\Delta = 8$	44.3567	45.1976	43.7985
PSNR at $\Delta = 12$	41.2904	42.1089	40.3840
PSNR at $\Delta = 14$	40.1984	40.9023	39.0461
PSNR at $\Delta = 16$	39.2341	39.8951	37.9381
Watermark size 32 × 32			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
PSNR at $\Delta = 8$	44.2742	44.5318	43.8941
PSNR at $\Delta = 12$	41.0020	41.1989	40.3535
PSNR at $\Delta = 14$	39.7006	39.8931	39.0123
PSNR at $\Delta = 16$	38.6869	38.9122	37.9012

Table 3-14 SSIM for different grey-scale images with different embedding strengths and watermark sizes

Watermark size 96 × 64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
SSIM at $\Delta = 8$	0.9850	0.9810	0.9930
SSIM at $\Delta = 12$	0.9728	0.9671	0.9805
SSIM at $\Delta = 14$	0.9601	0.9597	0.9799
SSIM at $\Delta = 16$	0.9492	0.9501	0.9880
Watermark size 64 × 64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
SSIM at $\Delta = 8$	0.9845	0.9805	0.9922
SSIM at $\Delta = 12$	0.9711	0.9676	0.9807
SSIM at $\Delta = 14$	0.9621	0.9599	0.9790
SSIM at $\Delta = 16$	0.9504	0.9507	0.9883
Watermark size 32 × 32			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
SSIM at $\Delta = 8$	0.9848	0.9812	0.9923
SSIM at $\Delta = 12$	0.9692	0.9678	0.9812
SSIM at $\Delta = 14$	0.9602	0.9478	0.9795
SSIM at $\Delta = 16$	0.9455	0.9387	0.9879


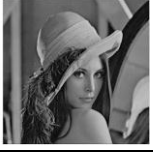

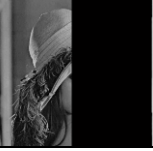


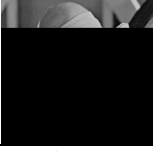






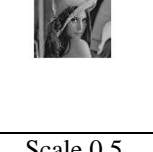













Table 3-15 Original and watermarked Lena images at different embedding strengths

	
Original Unwatermarked Lena image	
	
Watermarked image at $\Delta = 8$	Watermarked image at $\Delta = 12$
	
Watermarked image at $\Delta = 14$	Watermarked image at $\Delta = 16$

Table 3-16 Normalized correlation for Lena grey-scale image

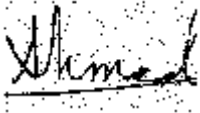
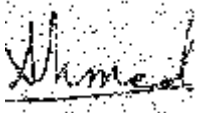
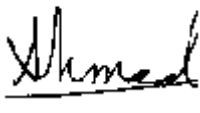
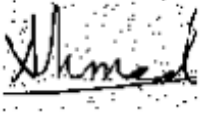


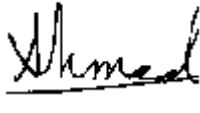

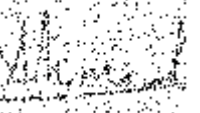
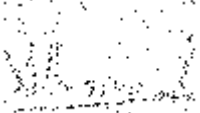


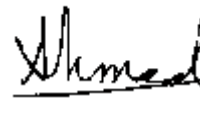
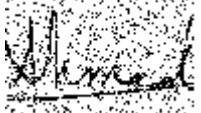




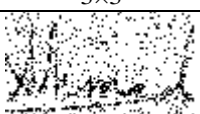
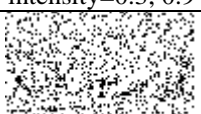
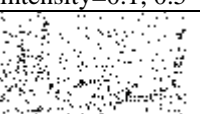





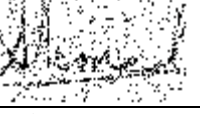
Watermark size 96×64, at $\Delta = 14$.			
Attacks	NC	Attacks	NC
Cropping 75% V	0.9646	Low pass 3×3	0.9716
Cropping 50% V	0.9701	Wiener 3×3	0.9911
Cropping 75% H	1	Median 3×3	0.9841
Cropping 75% V+ H	0.9834	JPEG 75	1
Scale 2	1	JPEG 50	1
Scale 0.75	0.9702	JPEG 30	0.9546
Scale 0.5	0.8846	Gaussian noise m=0, v=0.002	0.6289
S&P noise, d=0.02	0.7233	Gaussian noise m=0, v=0.001	0.7715
Contrast enhancements intensity=0.3, 0.9	0.9818	S&P noise, d=0.02+ Median 3×3	0.9897
Contrast enhancements intensity=0.1, 0.5	0.8842	S&P noise, d=0.05+ Median 3×3	0.9801
Stirmark_AFFINE_1	0.8985	Stirmark_CONV_1	0.9500
Stirmark_AFFINE_8	0.8199	Stirmark_RML_10	0.9615
Stirmark_ROTSCALE_0.25	0.9057	Stirmark_RML_50	0.9903
Stirmark_ROTSCALE_-0.5	0.8204	Stirmark_RML_100	0.9951
Stirmark_ROT_-0.5	0.8155	Stirmark_ROT_0.25	0.9095
Stirmark_ROTROP_-0.5	0.8170	Stirmark_ROTROP_0.25	0.9188
Watermark size 64×64, at $\Delta = 14$.			
Attacks	NC	Attacks	NC
Cropping 75% V	0.9754	Low pass 3×3	0.9812
Cropping 50% V	0.9789	Wiener 3×3	0.9902
Cropping 75% H	1	Median 3×3	0.9912
Cropping 75% V+ H	0.9841	JPEG 75	1
Scale 2	1	JPEG 50	1
Scale 0.75	0.9855	JPEG 30	0.9811
Scale 0.5	0.9430	Gaussian noise m=0, v=0.002	0.6823
S&P noise, d=0.02	0.8734	Gaussian noise m=0, v=0.001	0.8545
Contrast enhancements intensity=0.3, 0.9	0.9879	S&P noise, d=0.02+ Median 3×3	0.9910
Contrast enhancements intensity=0.1, 0.5	0.9423	S&P noise, d=0.05+ Median 3×3	0.9899
Stirmark_AFFINE_1	0.9575	Stirmark_CONV_1	0.9699
Stirmark_AFFINE_8	0.9070	Stirmark_RML_10	0.9754
Stirmark_ROTSCALE_0.25	0.9547	Stirmark_RML_50	0.9793
Stirmark_ROTSCALE_-0.5	0.9062	Stirmark_RML_100	0.9879
Stirmark_ROT_-0.5	0.9037	Stirmark_ROT_0.25	0.9680
Stirmark_ROTROP_-0.5	0.9051	Stirmark_ROTROP_0.25	0.9650
Watermark size 32×32, at $\Delta = 14$.			
Attacks	NC	Attacks	NC
Cropping 75% V	0.9879	Low pass 3×3	1
Cropping 50% V	0.9921	Wiener 3×3	1
Cropping 75% H	1	Median 3×3	1
Cropping 75% V+ H	0.9891	JPEG 75	1
Scale 2	1	JPEG 50	1
Scale 0.75	1	JPEG 30	1
Scale 0.5	0.9810	Gaussian noise m=0, v=0.002	0.6921
S&P noise, d=0.02	0.9521	Gaussian noise m=0, v=0.001	0.9530
Contrast enhancements intensity=0.3, 0.9	1	S&P noise, d=0.02+ Median 3×3	1
Contrast enhancements intensity=0.1, 0.5	0.9800	S&P noise, d=0.05+ Median 3×3	1
Stirmark_AFFINE_1	0.9913	Stirmark_CONV_1	1
Stirmark_AFFINE_8	0.9770	Stirmark_RML_10	1
Stirmark_ROTSCALE_0.25	0.9989	Stirmark_RML_50	1
Stirmark_ROTSCALE_-0.5	0.9683	Stirmark_RML_100	1
Stirmark_ROT_-0.5	0.9607	Stirmark_ROT_0.25	0.9978
Stirmark_ROTROP_-0.5	0.9727	Stirmark_ROTROP_0.25	0.9967

Table 3-17 Watermarked images after attacks

		
Cropping 75% V	Low pass 3×3	JPEG 75
		
Cropping 50% V	Wiener 3×3	JPEG 50
		
Cropping 75% H	Median 3×3	JPEG 30
		
Cropping 75% V+ H	S&P noise, $d=0.02$	Gaussian noise $m=0, v=0.002$
		
Scale 2	Scale 0.5	Gaussian noise $m=0, v=0.001$
		
S&P noise, $d=0.05+$ Median 3×3	Contrast enhancements intensity=0.3, 0.9	Contrast enhancements intensity=0.1, 0.5
		
Stirmark_AFFINE_1	Stirmark_AFFINE_8	Stirmark_CONV_1
		
Stirmark_ROTSCALE_-0.5	Stirmark_ROTSCALE_0.25	Stirmark_ROT_0.25
		
Stirmark_ROT_-0.5	Stirmark_ROTROP_-0.5	Stirmark_ROTROP_0.25

		
Stirmark_RML_10	Stirmark_RML_50	Stirmark_RML_100

Table 3-18 Extracted watermarks after attacks at watermark embedding strength $\Delta = 14$

		
Cropping 75% V	Low pass 3x3	JPEG 75
		
Cropping 50% V	Wiener 3x3	JPEG 50
		
Cropping 75% H	Median 3x3	JPEG 30
		
Cropping 75% V+H	S&P noise, $d=0.02$	Gaussian noise $m=0, v=0.002$
		
Scale 2	Scale 0.5	Gaussian noise $m=0, v=0.001$
		
S&P noise, $d=0.05+$ Median 3x3	Contrast enhancements intensity=0.3, 0.9	Contrast enhancements intensity=0.1, 0.5
		
Stirmark_AFFINE_1	Stirmark_AFFINE_8	Stirmark_CONV_1
		
Stirmark_ROTSCALE_-0.5	Stirmark_ROTSCALE_0.25	Stirmark_ROT_0.25
		
Stirmark_ROT_-0.5	Stirmark_ROT_CROP_-0.5	Stirmark_ROT_CROP_0.25

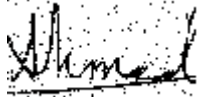
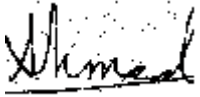
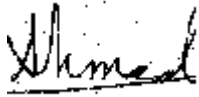


		
Stirmark_RML_10	Stirmark_RML_50	Stirmark_RML_100

Table 3-19 Extracted watermarks after attacks at watermark embedding strength $\Delta = 16$

		
Cropping 75% V	Low pass 3×3	JPEG 75
		
Cropping 50% V	Wiener 3×3	JPEG 50
		
Cropping 75% H	Median 3×3	JPEG 30
		
Cropping 75% V+H	S&P noise, $d=0.02$	Gaussian noise $m=0, v=0.002$
		
Scale 2	Scale 0.5	Gaussian noise $m=0, v=0.001$
		
S&P noise, $d=0.05+$ Median 3×3	Contrast enhancements intensity=0.3, 0.9	Contrast enhancements intensity=0.1, 0.5
		
Stirmark_AFFINE_1	Stirmark_AFFINE_8	Stirmark_CONV_1
		
Stirmark_ROTSCALE_-0.5	Stirmark_ROTSCALE_0.25	Stirmark_ROT_0.25
		
Stirmark_ROT_-0.5	Stirmark_ROTSCROP_-0.5	Stirmark_ROTSCROP_0.25

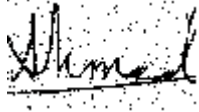
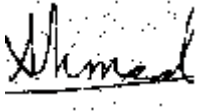
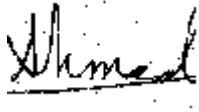



		
Stirmark_RML_10	Stirmark_RML_50	Stirmark_RML_100

Table 3-20 Matlab execution time

Intel processor centrino 2 GHZ, RAM= 1 GB			
<i>Watermark size</i>	<i>Embedding time</i>	<i>Extraction time</i>	<i>Total time</i>
96×64	3.25 seconds	1.53 seconds	4.78 seconds
64×64	3.25 seconds	1.46 seconds	4.71 seconds
32×32	3.20 seconds	3.18 seconds	6.38 seconds

Table 3-21 Performance evaluation against high resolution images

Watermark size 96×64					
					
(a): Original Host image		(b): Watermarked host image of size 1024×1024		(c): Watermarked host image of size 2048×2048	
PSNR	SSIM		PSNR	SSIM	
34.4	0.9722		28.98	0. 0.9702	
Attacks	host image size 1024×1024		host image size 2048×2048		
JPEG 30	NC= 0.9993		NC= 1.0000		
Cropping 75 % both sides	NC=0.9726		NC=0.9711		
Low-pass Filter 3×3	NC=0.9996		NC=1.0000		
Median filter 3×3	NC=0.9997		NC=1.0000		
Wiener filter 3×3	NC=1.0000		NC=1.0000		

3.6 Algorithm 3: Blind Image Watermarking for High capacity watermarks Using Low-Frequency Band DCT Coefficients

In this algorithm an image authentication technique evaluating the number of bits that can be embedded into a host image is proposed. The previous algorithms used only 8 coefficients of the 8×8 DCT block. Large watermark size such as 224×128 cannot be multiply embedded in algorithms 1 and 2. In order to solve this capacity problem, algorithm 3 uses 16 coefficients and can embed information up to 25% of the host image size, and this in turn increases the capacity of algorithm 3 over algorithms 1 and 2.

3.6.1 The Proposed High Capacity Image Watermarking Algorithm

Inside each 8×8 sub-blocks, 16 DCT coefficients are identified as shown in Figure 3.10. Each block of the DCT coefficients is subjected to a process of zigzag. The zigzag process progresses from low-frequency to high-frequency terms, where the first sixteen low frequencies excluding the DC coefficient will be selected. The bit embedding formula is defined in equation 3.5. The shuffle scheme is applied for each copy before embedding. The watermarked host image is obtained using the inverse DCT transform. Figure 3.11 shows all the embedding steps. The embedded watermarks information can be extracted by performing 8×8 DCT transform for the watermarked host image. The zigzag process will be applied again to indicate the same coefficients of the host image that carries the 16 bits of the embedded watermarks. The same secret key in the initial scrambling operation will be used. The watermark information can be retrieved by using a reverse process to the shuffle scheme which has been applied in the embedding process. The bit extraction formula is defined in equation 3.8. Figure 3.12 shows all the extraction steps.

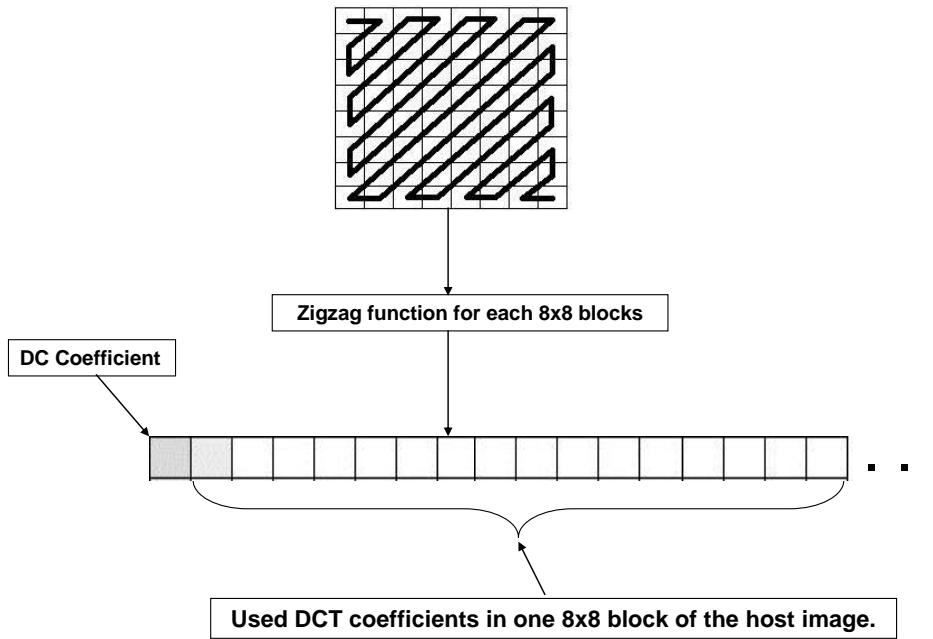


Figure 3-10 The used DCT coefficients in 8x8 sub-block of the host image.

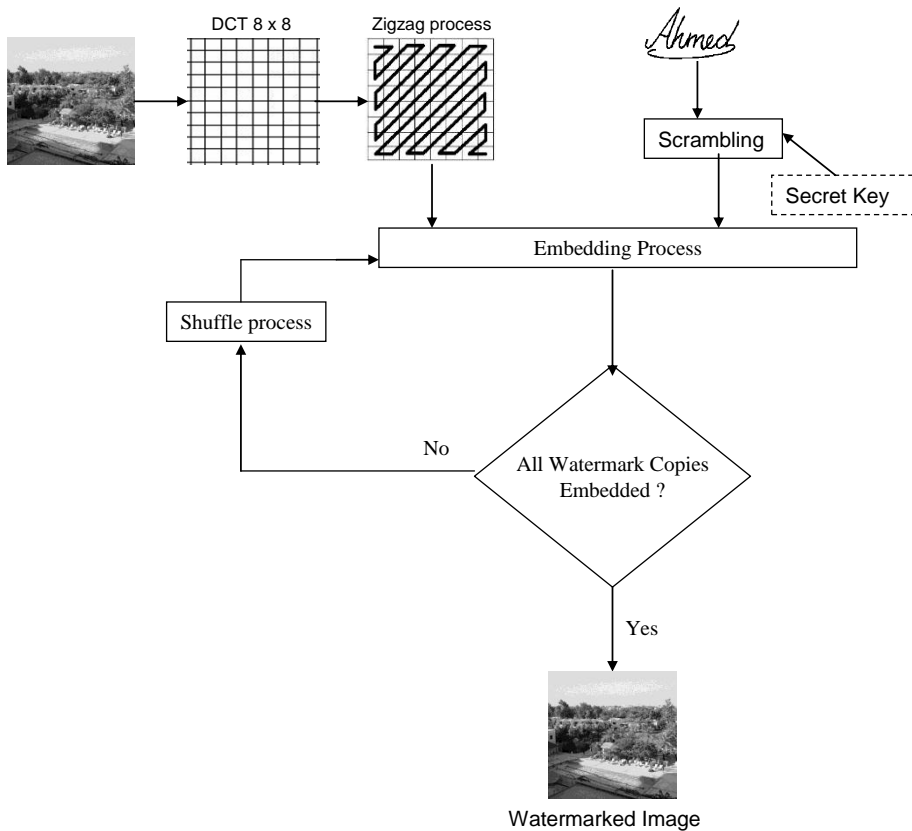


Figure 3-11 A flow graph for the embedding process.

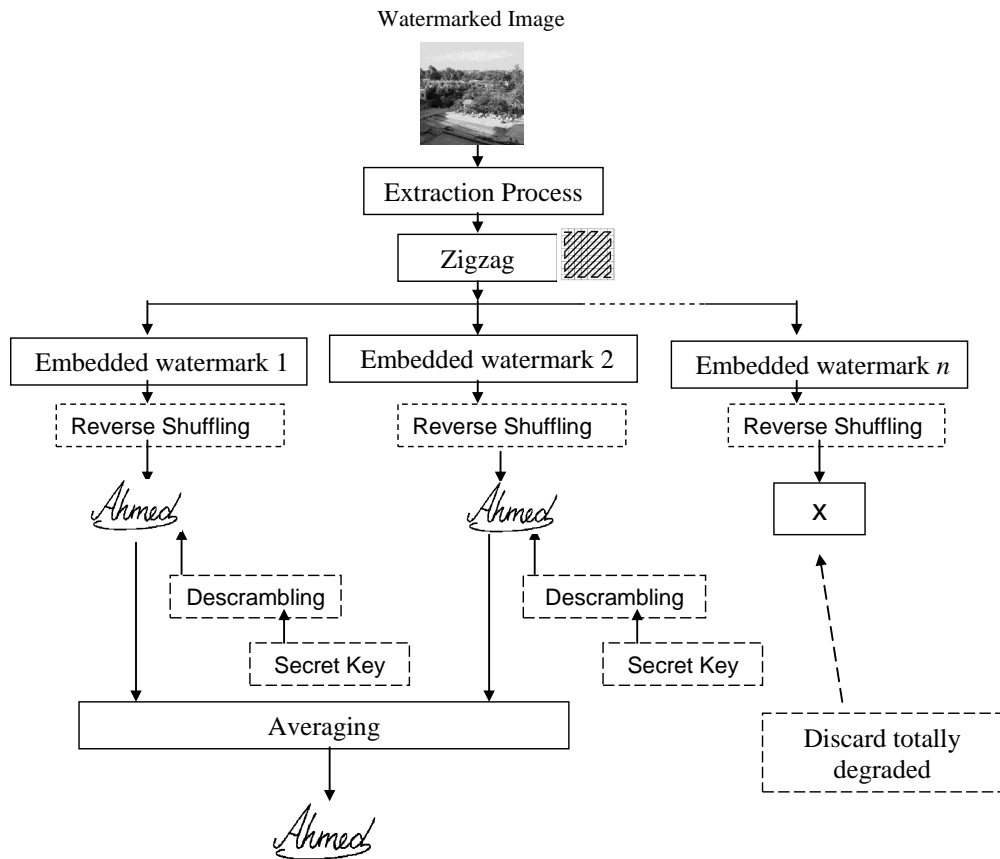


Figure 3-12 A flow graph for the extraction process.

3.6.2 Results

This algorithm is examined using different images of size 512×512 . Hand written signature images of sizes 224×128 and 192×64 are used as watermarks. Tables 3.22 and 3.23 demonstrate the perceptual invisibility of the proposed algorithm at different embedding strengths Δ and different watermark sizes. The PSNR values between the watermarked and original images are 41.7 dB and 36.9 dB for watermarking strengths $\Delta = 8$ and $\Delta = 16$, respectively. In Table 3.23 the perceptual invisibility of the proposed algorithm is evaluated using SSIM at different embedding strengths. The SSIM values between the watermarked and original images are 0.9724 and 0.9291 for watermarking strengths $\Delta = 8$ and $\Delta = 16$, respectively.

Various embedding strengths have been investigated to determine which values provide the best performance for the majority of the images. It was found that $\Delta = 14$ is the best compromise between the robustness and the distortion introduced to the watermarked image. Higher embedding strength values such as $\Delta = 16$ will provide stronger robustness, but it will introduce more distortion in the host images.

Different attacks are applied to the watermarked images. As shown in Table 3.25, NC is used to measure the similarity between the original and the extracted watermarks. A visual inspection of the extracted watermarks is depicted in Tables 3.27 and 3.28 at watermarking strengths $\Delta = 14$ and $\Delta = 16$, respectively.

The execution time of the Matlab algorithms on a 2 Ghz Centrino processor and 1 Gb is shown in Table 3.29. Finally, the performance evaluation against high resolution images is illustrated in Table 3.30.

Table 3-22 PSNR for different grey-scale images with different embedding strengths and watermark sizes

Watermark size 224×128			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
PSNR at $\Delta = 8$	41.7302	42.7355	40.8218
PSNR at $\Delta = 12$	38.9117	39.9296	37.4441
PSNR at $\Delta = 14$	37.8140	38.8259	36.1740
PSNR at $\Delta = 16$	36.9027	37.8645	35.1072
Watermark size 192×64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
PSNR at $\Delta = 8$	41.7235	42.7123	40.8410
PSNR at $\Delta = 12$	38.9037	39.9178	37.4249
PSNR at $\Delta = 14$	37.8112	38.8217	36.1534
PSNR at $\Delta = 16$	36.9021	37.8534	35.1012
Watermark size 128×64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
PSNR at $\Delta = 8$	41.7176	42.7120	40.8338
PSNR at $\Delta = 12$	38.9033	39.9167	37.4240
PSNR at $\Delta = 14$	37.8109	38.8216	36.1523
PSNR at $\Delta = 16$	36.9019	37.8531	35.1010

Table 3-23 SSIM for different grey-scale images with different embedding strengths and watermark sizes

Watermark size 224 × 128			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
SSIM at $\Delta = 8$	0.9724	0.9708	0.9876
SSIM at $\Delta = 12$	0.9521	0.9478	0.9749
SSIM at $\Delta = 14$	0.9405	0.9349	0.9673
SSIM at $\Delta = 16$	0.9291	0.9205	0.9597
Watermark size 192 × 64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
SSIM at $\Delta = 8$	0.9722	0.9708	0.9877
SSIM at $\Delta = 12$	0.9519	0.9477	0.9748
SSIM at $\Delta = 14$	0.9405	0.9345	0.9670
SSIM at $\Delta = 16$	0.9289	0.9205	0.9596
Watermark size 128 × 64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
SSIM at $\Delta = 8$	0.9721	0.9706	0.9876
SSIM at $\Delta = 12$	0.9520	0.9477	0.9747
SSIM at $\Delta = 14$	0.9403	0.9347	0.9672
SSIM at $\Delta = 16$	0.9290	0.9201	0.9595

Table 3-24 Original and watermarked Lena images at different embedding strengths



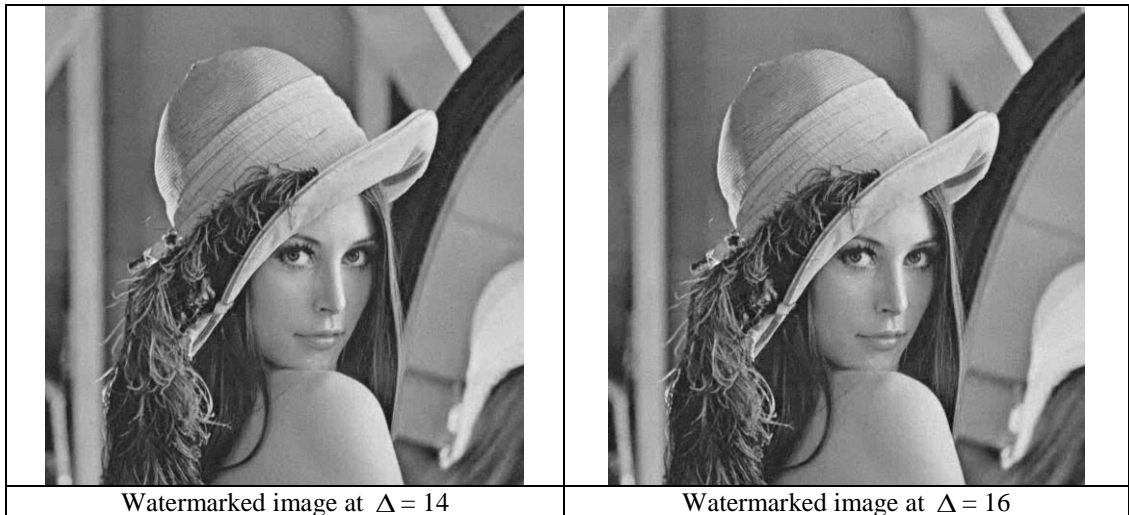







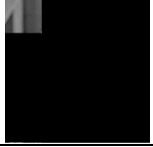









Table 3-25 Normalized correlation for Lena grey-scale image

Watermark size 224×128 , at $\Delta = 14$.			
Attacks	NC	Attacks	NC
Cropping 75% V	0.9834	Low pass 3×3	0.9853
Cropping 50% V	0.9838	Wiener 3×3	0.9937
Cropping 75% H	0.9918	Median 3×3	0.9925
Cropping 75% V+H	0.9943	JPEG 75	0.9998
Scale 2	1	JPEG 50	0.9991
Scale 0.75	0.9713	JPEG 30	0.9879
Scale 0.5	0.9279	Gaussian noise $m=0, v=0.002$	0.7760
S&P noise, $d=0.02$	0.8665	Gaussian noise $m=0, v=0.001$	0.8574
Contrast enhancements intensity=0.3, 0.9	0.9940	S&P noise, $d=0.02+$ Median 3×3	0.9910
Contrast enhancements intensity=0.1, 0.5	0.9722	S&P noise, $d=0.05+$ Median 3×3	0.9867
StirMark_AFFINE_1	0.9565	StirMark_CONV_1	0.9777
StirMark_AFFINE_8	0.9268	StirMark_RML_10	0.9667
StirMark_ROTSCALE_0.25	0.9501	StirMark_RML_50	0.9879
StirMark_ROTSCALE_-0.5	0.9221	StirMark_RML_100	0.9905
StirMark_ROT_-0.5	0.9223	StirMark_ROT_0.25	0.9503
StirMark_ROTROP_-0.5	0.9215	StirMark_ROTROP_0.25	0.9547
Watermark size 192×64 , at $\Delta = 14$.			
Attacks	NC	Attacks	NC
Cropping 75% V	0.9664	Low pass 3×3	0.9831
Cropping 50% V	0.9742	Wiener 3×3	0.9950
Cropping 75% H	1	Median 3×3	0.9930
Cropping 75% V+H	0.9873	JPEG 75	1
Scale 2	1	JPEG 50	0.9990
Scale 0.75	0.9604	JPEG 35	0.9956
Scale 0.5	0.8989	Gaussian noise $m=0, v=0.002$	0.5493
S&P noise, $d=0.02$	0.7372	Gaussian noise $m=0, v=0.001$	0.7224
Contrast enhancements intensity=0.3, 0.9	0.9917	S&P noise, $d=0.02+$ Median 3×3	0.9921
Contrast enhancements intensity=0.1, 0.5	0.9504	S&P noise, $d=0.05+$ Median 3×3	0.9873
StirMark_AFFINE_1	0.8874	StirMark_CONV_1	0.9342
StirMark_AFFINE_8	0.8157	StirMark_RML_10	0.9022
StirMark_ROTSCALE_0.25	0.8640	StirMark_RML_50	0.9575
StirMark_ROTSCALE_-0.5	0.8087	StirMark_RML_100	0.9684
StirMark_ROT_-0.5	0.8086	StirMark_ROT_0.25	0.8690
StirMark_ROTROP_-0.5	0.8078	StirMark_ROTROP_0.25	0.8794

Watermark size 128 × 64, at $\Delta = 14$.			
Attacks	NC	Attacks	NC
Cropping 75% V	0.9699	Low pass 3×3	0.9857
Cropping 50% V	0.9787	Wiener 3×3	0.9974
Cropping 75% H	1	Median 3×3	0.9960
Cropping 75% V+H	0.9893	JPEG 75	1
Scale 2	1	JPEG 50	0.9996
Scale 0.75	0.9751	JPEG 35	0.9975
Scale 0.5	0.9329	Gaussian noise m=0, v=0.002	0.6915
S&P noise, d=0.02	0.8759	Gaussian noise m=0, v=0.001	0.8642
Contrast enhancements intensity=0.3, 0.9	0.9959	S&P noise, d=0.02+ Median 3×3	0.9952
Contrast enhancements intensity=0.1, 0.5	0.9679	S&P noise, d=0.05+ Median 3×3	0.9931
Stirmark_AFFINE_1	0.9346	Stirmark_CONV_1	0.9696
Stirmark_AFFINE_8	0.8800	Stirmark_RML_10	0.9622
Stirmark_ROTSCALE_0.25	0.9278	Stirmark_RML_50	0.9903
Stirmark_ROTSCALE_-0.5	0.8699	Stirmark_RML_100	0.9956
Stirmark_ROT_-0.5	0.8677	Stirmark_ROT_0.25	0.9286
Stirmark_ROTROP_-0.5	0.8738	Stirmark_ROTROP_0.25	0.9359

Table 3-26 Watermarked images after attacks

		
Cropping 75% V	Low pass 3×3	JPEG 75
		
Cropping 50% V	Wiener 3×3	JPEG 50
		
Cropping 75% H	Median 3×3	JPEG 30
		
Cropping 75% V+H	S&P noise, d=0.02	Gaussian noise m=0, v=0.002
		
Scale 2	Scale 0.5	Gaussian noise m=0, v=0.001
		
S&P noise, d=0.05+ Median 3×3	Contrast enhancements intensity=0.3, 0.9	Contrast enhancements intensity=0.1, 0.5

Stirmark_AFFINE_1	Stirmark_AFFINE_8	Stirmark_CONV_1	
Stirmark_ROTSCALE_-0.5	Stirmark_ROTSCALE_0.25	Stirmark_ROT_0.25	
Stirmark_ROT_-0.5	Stirmark_ROTSCROP_-0.5	Stirmark_ROTSCROP_0.25	
Stirmark_RML_10	Stirmark_RML_50	Stirmark_RML_100	

Table 3-27 Extracted watermarks after attacks at watermark embedding strength $\Delta = 14$

Cropping 75% V	Low pass 3x3	JPEG 75
Cropping 50% V	Wiener 3x3	JPEG 50
Cropping 75% H	Median 3x3	JPEG 30

Cropping 75% V+ H	S&P noise, $d=0.02$	Gaussian noise $m=0, v=0.002$
Scale 2	Scale 0.5	Gaussian noise $m=0, v=0.001$
S&P noise, $d=0.05$ + Median 3×3	Contrast enhancements intensity= $0.3, 0.9$	Contrast enhancements intensity= $0.1, 0.5$
Stirmark_AFFINE_1	Stirmark_AFFINE_8	Stirmark_CONV_1
Stirmark_ROTSCALE_-0.5	Stirmark_ROTSCALE_0.25	Stirmark_ROT_0.25
Stirmark_ROT_-0.5	Stirmark_ROTROP_-0.5	Stirmark_ROTROP_0.25




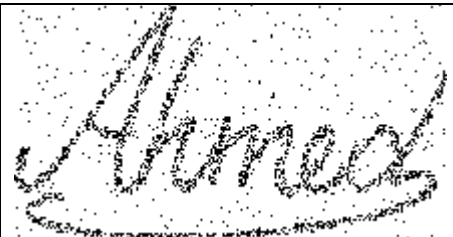
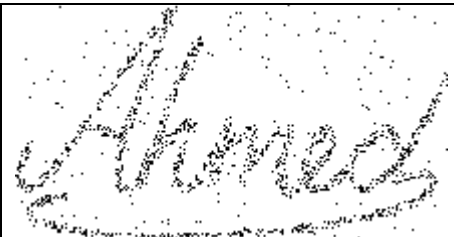


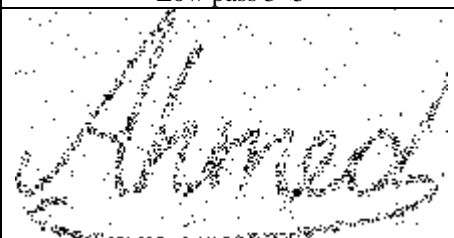
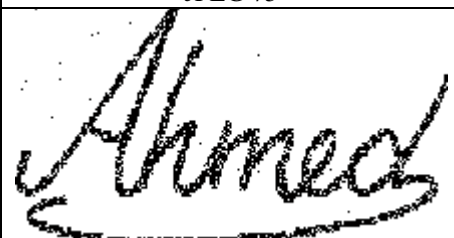
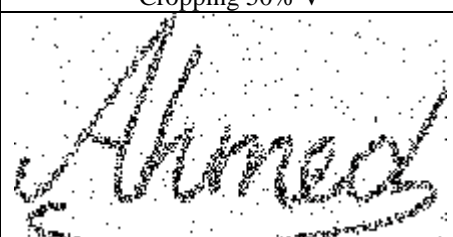
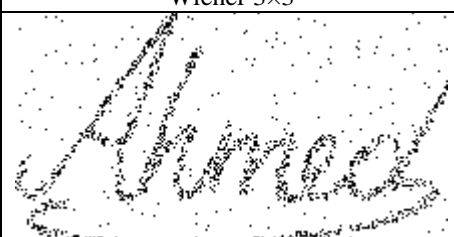
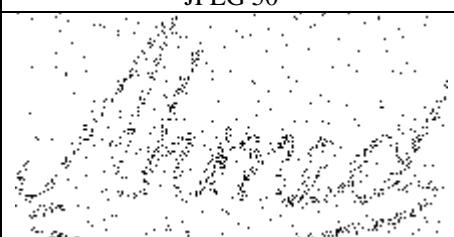
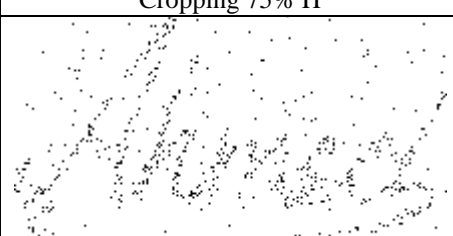
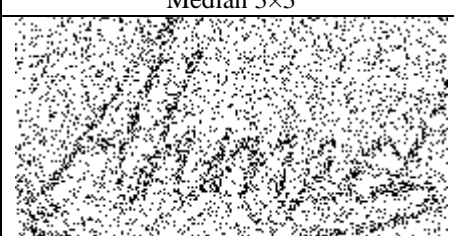
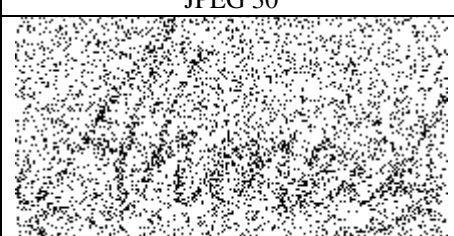
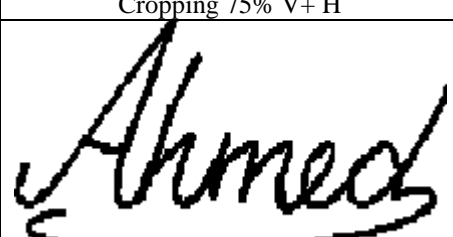
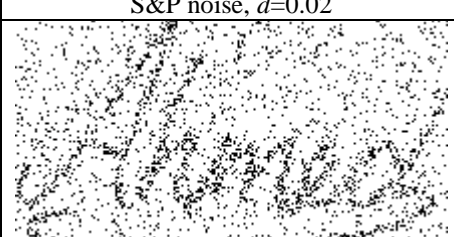
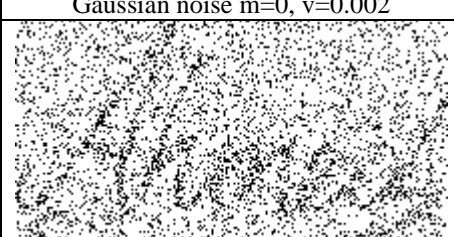
		
Stirmark_RML_10	Stirmark_RML_50	Stirmark_RML_100

Table 3-28 Extracted watermarks after attacks at watermark embedding strength $\Delta = 16$




		
Cropping 75% V	Low pass 3x3	JPEG 75
		
Cropping 50% V	Wiener 3x3	JPEG 50
		
Cropping 75% H	Median 3x3	JPEG 30
		
Cropping 75% V+H	S&P noise, $d=0.02$	Gaussian noise $m=0, v=0.002$
		
Scale 2	Scale 0.5	Gaussian noise $m=0, v=0.001$

S&P noise, $d=0.05+$ Median 3×3	Contrast enhancements intensity=0.3, 0.9	Contrast enhancements intensity=0.1, 0.5
Stirmark_AFFINE_1	Stirmark_AFFINE_8	Stirmark_CONV_1
Stirmark_ROTSCALE_-0.5	Stirmark_ROTSCALE_0.25	Stirmark_ROT_0.25
Stirmark_ROT_-0.5	Stirmark_ROTSCROP_-0.5	Stirmark_ROTSCROP_0.25
Stirmark_RML_10	Stirmark_RML_50	Stirmark_RML_100

Table 3-29 Matlab execution time

Intel processor centrino 2 GHZ, RAM= 1 GB			
Watermark size	Embedding time	Extraction time	Total time
224×128	4.78 seconds	7.82 seconds	12.60 seconds
224×96	4.78 seconds	5.51 seconds	10.29seconds
192×64	4.73 seconds	2.98 seconds	7.71 seconds

Table 3-30 Performance evaluation against high resolution images

Watermark size 224×128			
			
(a): Original Host image			
			
(b): Watermarked host image of size 1024×1024		(c): Watermarked host image of size 2048×2048	
PSNR	SSIM	PSNR	SSIM
31.7	0.9394	26.01	0.9352
Attacks	host image size 1024×1024	host image size 2048×2048	
JPEG 30	NC= 1.0000	NC= 1.0000	
Cropping 75 % both sides	NC=0.9144	NC=0.9986	
Low-pass Filter 3×3	NC=1.0000	NC=1.0000	
Median filter 3×3	NC=1.0000	NC=1.0000	
Wiener filter 3×3	NC=1.0000	NC=1.0000	

3.7 Comparison with previous work

Tables 3-31 and 3-32 represent comparisons between the proposed grey-scale algorithms (1, 2 and 3) and the watermarking method in [51, 54-57]. The algorithms introduced in [55] produce higher PSNR values than our proposed methods. However the method in [55] was tested against Low-pass, high-pass and JPEG attacks only, and from the results in Table 3-32 our methods outperform against attacks. Note that higher PSNR values can be achieved in our methods by changing the watermarking strength value. Also our algorithm is better than the proposed method in [57]. The watermark information was embedded in the DC component of each sub-block. Using the DC components for embedding limits the watermark size. For example, the maximum watermark size that can be embedded when using a host of 512×512 is 64×64 . It can be observed from 3-31 and 3-32 that the proposed algorithms generates higher PSNR and NC values and the extracted watermark is better compared to the remaining watermarking methods in Table 3-31.

Table 3-31 Comparison between proposed algorithms and others

Algorithm	Blind	Domain	Host size	Watermark size	PSNR
[51]	Yes	DCT	512×512	128×128	31.2
[54]	Yes	DCT	512×512	32×32	35.7
[55]	No	DCT	512×512	128×128	42.2
[56]	Yes	DCT	256×256	32×32	40.1
[57]	Yes	DCT	512×512	64×64	44.2
<i>Algorithm 1</i>	<i>Yes</i>	<i>DCT</i>	<i>512×512</i>	<i>96×64</i>	40.2
<i>Algorithm 2</i>	<i>Yes</i>	<i>DCT</i>	<i>512×512</i>	<i>96×64</i>	40.3
<i>Algorithm 3</i>	<i>Yes</i>	<i>DCT</i>	<i>512×512</i>	<i>224×128</i>	37.8

Table 3-32 Comparison of robustness

Attacks	Watermark size 128×128		Watermark size 128×128		
	Method in [51]		Algorithm 1	Algorithm 2	Algorithm 3
	NC		NC $\Delta=14$	NC $\Delta=14$	NC $\Delta=14$
Low-pass Filter 3×3	0.7511		0.9715	0.9717	0.9821
JPEG 80	0.9893		1	1	1
JPEG 60	0.8756		1	1	1
JPEG 40	0.7243		0.9913	0.9905	0.9956
Median filter 3×3	0.8132		0.9852	0.9846	0.9892
Scaling 0.75	0.5010		1	1	0.9986
Scaling 0.5	0.4441		0.9908	0.9901	0.9933

Watermark size 32×32		Watermark size 32×32		
	Method in [54]	Algorithm 1	Algorithm 2	Algorithm 3
Attacks	NC	NC $\Delta=14$	NC $\Delta=14$	NC $\Delta=14$
Low-pass Filter 3×3	0.95	1	1	1
Median filter 3×3	0.96	1	1	1
JPEG 50	0.99	1	1	1
Gaussian noise 0.001	0.97	0.9586	0.9530	0.9884
Histogram equalization	1	0.8075	0.8061	0.8751
Cropping 25%	0.74	1	1	1
Watermark size 128×128		Watermark size 128×128		
	Method in [55]	Algorithm 1	Algorithm 2	Algorithm 3
Attacks	NC	NC $\Delta=14$	NC $\Delta=14$	NC $\Delta=14$
No attack	0.9607	1	1	1
Low-pass filter	0.8607	0.9725	0.9717	0.9821
JPEG	0.8158	1	1	1
High-pass filter	0.7026	0.8527	0.8519	0.8974
Watermark size 32×32		Watermark size 32×32		
	Method in [56]	Algorithm 1	Algorithm 2	Algorithm 3
Attacks	NC	NC $\Delta=14$	NC $\Delta=14$	NC $\Delta=14$
Low-pass Filter 3×3	0.9855	1	1	1
Median filter 3×3	0.9499	1	1	1
Wiener filter 3×3	0.9711	1	1	1
JPEG 30	0.7911	1	1	1
Gaussian noise 0.003	0.9465	0.5988	0.5920	0.5679
Salt & Pepper 0.01	0.9543	0.9897	0.9866	1
Cropping 25%	0.9663	1	1	1
Scaling 0.5	0.9509	0.9832	0.9810	1
Scaling 0.8	0.9691	1	1	1
Scaling 1.5	0.9837	1	1	1
Watermark size 64×64		Watermark size 64×64		
	Method in [57]	Algorithm 1	Algorithm 2	Algorithm 3
Attacks	NC	NC $\Delta=14$	NC $\Delta=14$	NC $\Delta=14$
JPEG 65	1	1	1	1
JPEG 50	0.9596	1	1	1
Median filter 3×3	0.8913	0.9961	0.9912	1
S & P noise $d=0.001$	0.8869	0.9544	0.9521	1
Scale 0.75	0.6267	0.9875	0.9855	1
Gaussian noise 0.0005	0.8014	0.9860	0.9851	0.9966

3.8 Final Remarks

In this chapter, three different watermarking algorithms for grey-scale images have been developed. The developed watermarking algorithms utilized the DCT. The developed algorithms have been found to be robust against JPEG compression, cropping, small degrees of rotation, scaling, additive noise, filtering operations and Stirmark attacks. Handwritten signatures have been used as watermarks rather than the conventional pseudo random numbers. The use of such information as watermarks makes it possible to identify the owner by visual inspection of the extracted signatures. Single watermarks have been embedded multiple times in the host image. The developed techniques are blind. The shuffle scheme has been used to shield the watermark against cropping attacks.

For security reasons, a DCT coefficients selection process has been developed to increase the robustness, security of the proposed algorithm 2 and to reduce the visual changes when viewed by human eyes. . The location of the watermark in algorithm 1 was fixed (i.e. the first 8 AC coefficients) which compromised the security of the technique. In algorithm 2 the first 16 low frequency coefficients (excluding the DC value) in the 8×8 DCT block were screened and the eight coefficients with the maximum magnitudes were selected for embedding. The selected coefficients in algorithm 2 are adaptive and related to host image information.

The proposed algorithms 1 and 2 used only 8 coefficients of the 8×8 DCT block. Large watermark size such as 224×128 cannot be multiply embedded using algorithms 1 and 2. Algorithm 3 uses 16 coefficients and can embed information up to 25% of the host image size and this in turn increases the capacity of algorithm 3 over algorithms 1 and 2.

The fidelity of the proposed algorithms was examined using PSNR and SSIM measurements. The new algorithms performances were also compared to a DCT algorithm in [51, 54-57] to demonstrate their superiority. The new algorithms are used for grey-scale images. They will be extended to cover colour images in the next chapter.

Chapter 4 Digital Watermarking Algorithms for Colour Images

4.1 Overview

This chapter modifies the previous 3 algorithms of chapter 3 to work with coloured images. Two algorithms are presented in this chapter which embed the watermarks in the Y and G components of the colour image. Finally, a third algorithm with higher capacity is discussed.

4.2 Algorithm 4: Watermarking of Colour Images in the DCT Domain Using the Y Channel

Algorithm 4 is the coloured version of algorithm 2 of chapter 3. The watermark is embedded into the host images by selectively modifying the very low frequency parts of the DCT transformation. The eight coefficients in each 8×8 sub block can be defined using the DCS process. The watermark image is scrambled by a secret key and it is then converted to a vector. The binary watermark is shuffled using the shuffle scheme.

4.2.1 Embedding and Extraction Algorithms

The colour images are generally built with three colour channels. In the technique presented here, the colour image is decomposed into three components: Red (R), Green (G) and Blue (B). Three other planes that are obtained: Y, Cr and Cb. The Y components corresponds to the luminance information while, (Cr and Cb) components represent colour information. Watermark information is embedded in the Y plane using equation 3.5. The resultant Y' plane includes the watermark information. The inverse of the new YCrCb planes is built to obtain the RGB image. All the previous watermarking steps are described graphically in the diagram shown in Figure 4.1.

The extraction process doesn't require the original unmarked image. The recovery function is simply the inverse of the embedding function as illustrated in Figure 4.2. The watermark can be retrieved by using the same secret key and performing the inverse of the shuffle scheme. It is important to note that the watermark is embedded several times in the host image (depending on the sizes of the host image and the watermarked image).

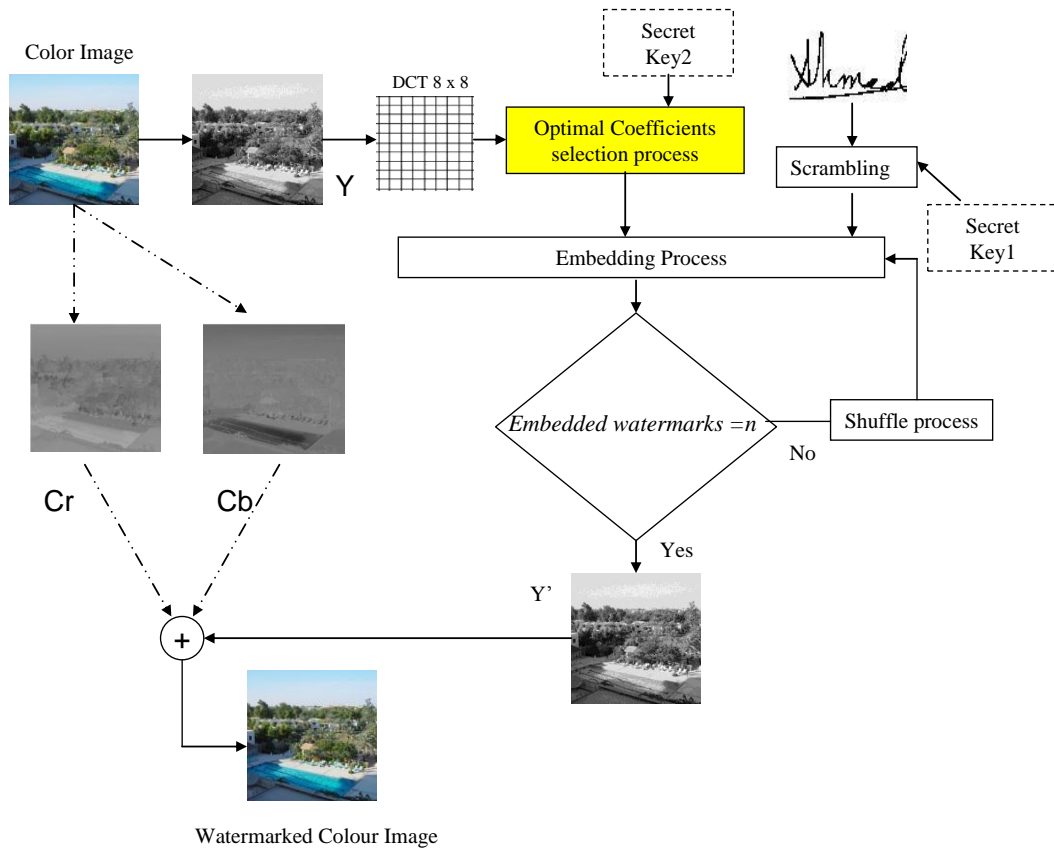


Figure 4-1 Graphical presentation for embedding steps

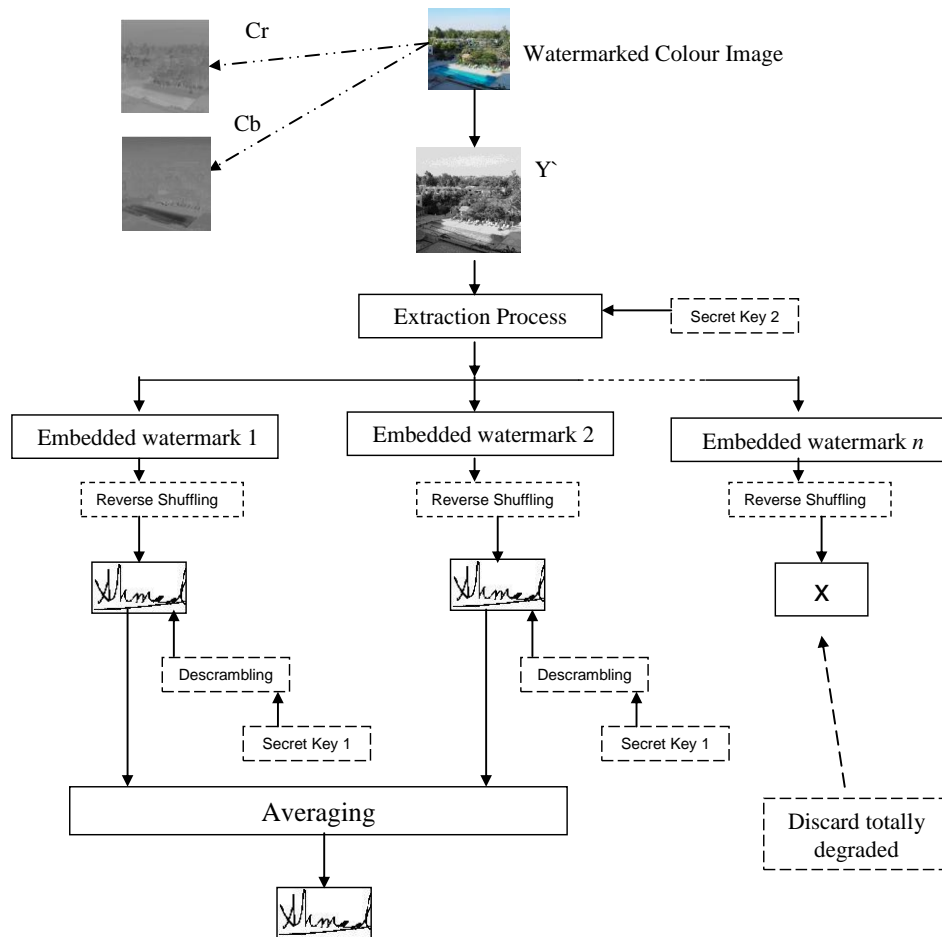


Figure 4-2 Graphical presentation for extraction steps

4.2.2 Simulation and Results

To confirm the invisibility of the embedding process several colour images of size 512×512 are used as a host image. Binary handwritten signatures of size 96×64 , 64×64 and 32×32 were used as the watermarks. To verify the robustness of the proposed method, various common signal processing and geometric attacks are applied to the watermarked images. As before, PSNR and SSIM are used to evaluate the performance of the proposed algorithm. Tables 4.1 and 4.2 demonstrate the perceptual invisibility of the proposed algorithm at different embedding strengths. It is worth noting that higher watermarking strength will provide strong robustness while maintaining the invisibility qualities. A visual comparison between the original Lena image and the watermarked versions at different embedding strengths are shown in Table 4.3.

Moreover, NC is used to measure the similarity between the original and the extracted watermark. It is worth mentioning that using smaller watermarks enables the user to embed more signatures in the host image. This will enable the watermark to survive different attacks and will produce higher NC values as shown in Table 4.4. A visual inspection of the extracted watermarks against different attacks is depicted in Tables 4.6 and 4.7 for $\Delta = 12$ and $\Delta = 16$, respectively.

The Matlab execution time on a 2Ghz Centrino processor and 1 Gb memory is shown in Table 4.8. Finally, the performance evaluation against high resolution images is illustrated in Table 4.9.

Table 4-1 PSNR for watermarked colour images

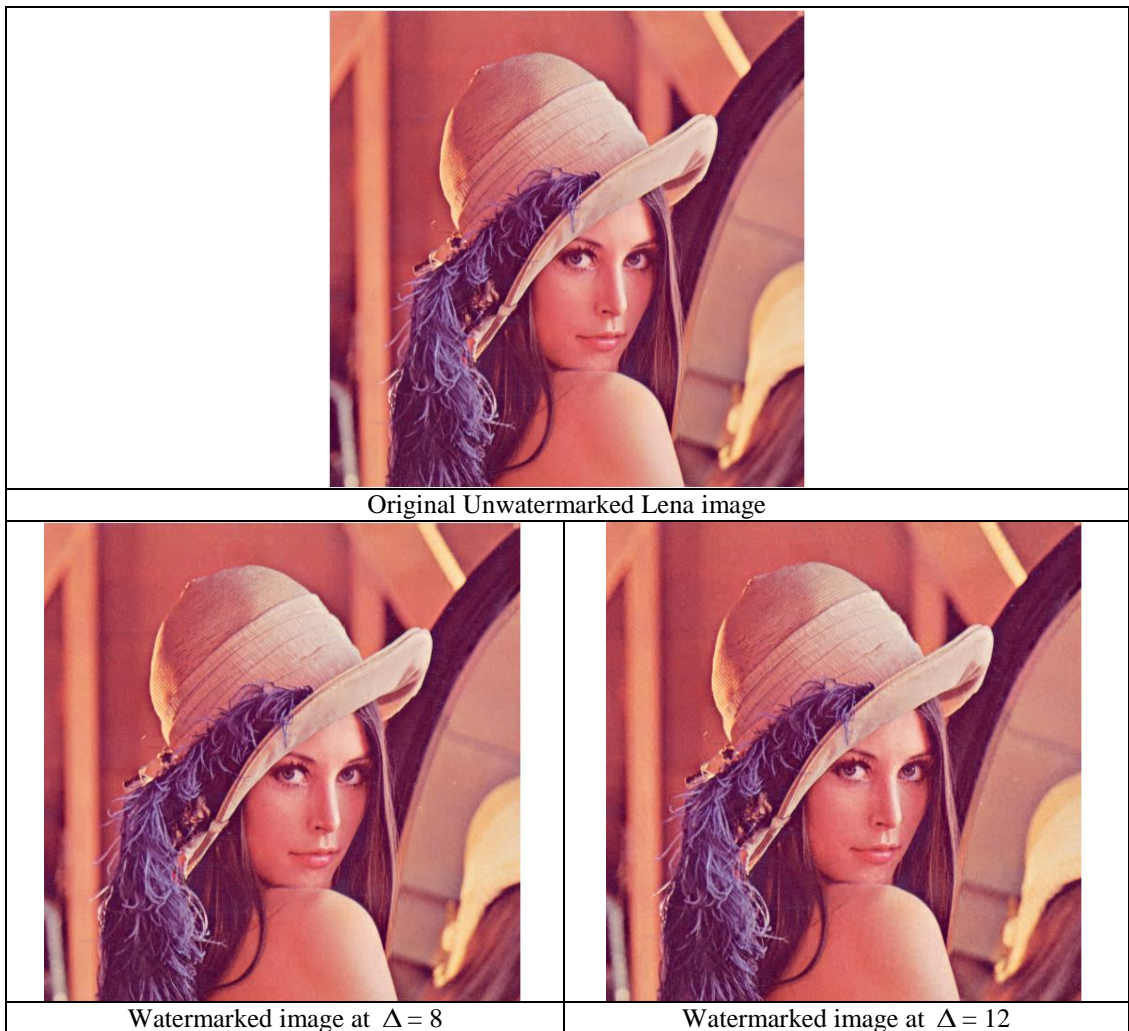
Watermark size 96 × 64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
PSNR at $\Delta = 8$	42.5692 dB	42.1671 dB	42.9671 dB
PSNR at $\Delta = 12$	39.9001 dB	39.7112 dB	40.1212 dB
PSNR at $\Delta = 16$	37.8383 dB	37.3123 dB	37.9412 dB
PSNR at $\Delta = 20$	36.2696 dB	36.1971 dB	36.9145 dB
PSNR at $\Delta = 24$	35.0019 dB	34.9017 dB	35.7412 dB
Watermark size 64 × 64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
PSNR at $\Delta = 8$	42.8193 dB	42.4671 dB	43.1095 dB
PSNR at $\Delta = 12$	40.3141 dB	39.9112 dB	40.5471 dB
PSNR at $\Delta = 16$	37.9313 dB	37.8123 dB	38.3617 dB
PSNR at $\Delta = 20$	36.7390 dB	36.7971 dB	37.2391 dB
PSNR at $\Delta = 24$	35.4027 dB	35.0114 dB	35.9102 dB
Watermark size 32 × 32			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
PSNR at $\Delta = 8$	42.9713 dB	42.6801 dB	43.3196 dB
PSNR at $\Delta = 12$	40.5082 dB	40.2156 dB	40.7451 dB
PSNR at $\Delta = 16$	38.0112 dB	37.9913 dB	38.6581 dB
PSNR at $\Delta = 20$	36.9850 dB	36.9951 dB	37.5437 dB
PSNR at $\Delta = 24$	35.7422 dB	35.3734 dB	36.1130 dB

Table 4-2 SSIM for watermarked colour images

Watermark size 96 × 64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
SSIM at $\Delta = 8$	0.9776	0.9709	0.9916
SSIM at $\Delta = 12$	0.9595	0.9442	0.9834
SSIM at $\Delta = 16$	0.9365	0.9123	0.9734
SSIM at $\Delta = 20$	0.9095	0.8776	0.9617
SSIM at $\Delta = 24$	0.8789	0.8386	0.9494

Watermark size 64 × 64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
SSIM at $\Delta = 8$	0.9792	0.9753	0.9917
SSIM at $\Delta = 12$	0.9646	0.9566	0.9837
SSIM at $\Delta = 16$	0.9467	0.9357	0.9743
SSIM at $\Delta = 20$	0.9281	0.9110	0.9641
SSIM at $\Delta = 24$	0.9070	0.8838	0.9529
Watermark size 32 × 32			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
SSIM at $\Delta = 8$	0.9801	0.9765	0.9920
SSIM at $\Delta = 12$	0.9661	0.9600	0.9840
SSIM at $\Delta = 16$	0.9483	0.9367	0.9744
SSIM at $\Delta = 20$	0.9293	0.9124	0.9644
SSIM at $\Delta = 24$	0.9090	0.8891	0.9538

Table 4-3 Original and watermarked Lena images at different embedding strengths



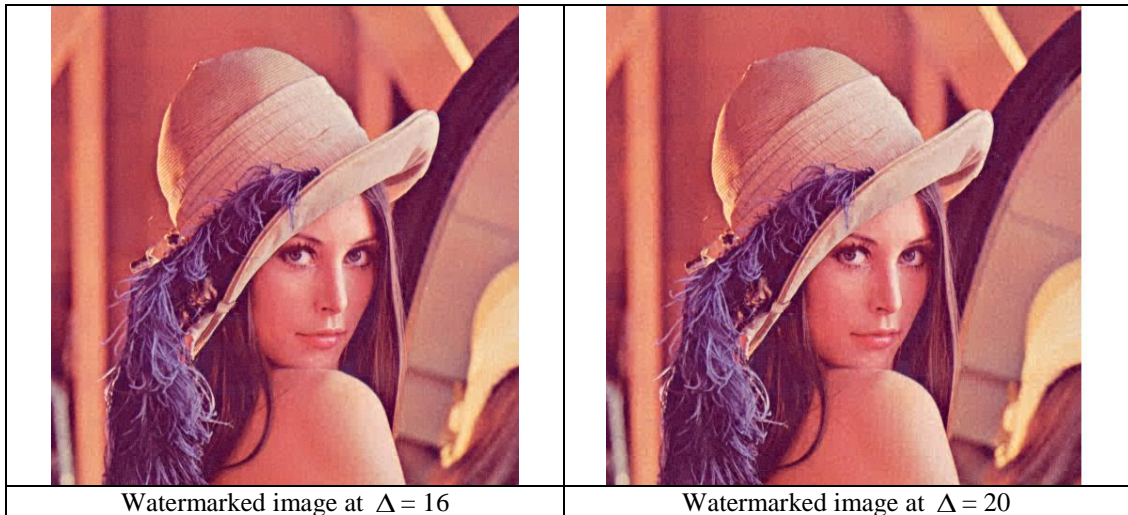
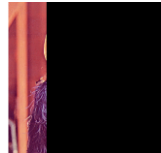












Table 4-4 Normalized correlation for Lena colour image

Watermark size 96×64, at $\Delta = 12$.			
Attacks	NC	Attacks	NC
Cropping 75% V	0.9912	Low pass 3×3	0.9801
Cropping 50% V	0.9991	Wiener 3×3	0.9931
Cropping 75% H	1	Median 3×3	0.9920
Cropping 75% V+ H	0.9943	JPEG 75	1
Scale 0.75	0.9735	JPEG 50	1
Scale 0.5	0.8800	JPEG 40	0.9973
Gauss. noise m=0,v=0.002	0.7801	JPEG 30	0.9640
S&P noise, d=0.02 + Median 3×3	0.9925	S&P noise, d=0.01	0.8715
Contrast enhancements intensity=0.3, 0.9	0.9851	Scale 2	1
Stirmark_AFFINE_1	0.9024	Stirmark_CONV_1	0.9451
Stirmark_AFFINE_8	0.8155	Stirmark_RML_10	0.9538
Stirmark_ROTSCALE_0.25	0.8985	Stirmark_RML_100	0.9929
Stirmark_ROTSCALE_-0.5	0.8100	Stirmark_SS_1	0.9630
Stirmark_ROT_0.25	0.9042	Stirmark_SS_2	1
Stirmark_ROT_-0.5	0.8100	Stirmark_SS_3	0.9093
Stirmark_ROTROP_-0.5	0.8073	Stirmark_ROTROP_0.25	0.9086
Watermark size 64×64, at $\Delta = 12$.			
Attacks	NC	Attacks	NC
Cropping 75% V	0.9958	Low pass 3×3	0.9862
Cropping 50% V	0.9964	Wiener 3×3	0.9991
Cropping 75% H	1	Median 3×3	0.9942
Cropping 75% V+ H	0.9947	JPEG 75	1
Scale 0.75	0.9834	JPEG 50	1
Scale 0.5	0.9471	JPEG 40	0.9989
Gauss. noise m=0,v=0.002	0.9001	JPEG 30	0.9872
S&P noise, d=0.02 + Median 3×3	0.9938	S&P noise, d=0.01	0.9576
Contrast enhancements intensity=0.3, 0.9	0.9962	Scale 2	1
Stirmark_AFFINE_1	0.9619	Stirmark_CONV_1	0.9749
Stirmark_AFFINE_8	0.9065	Stirmark_RML_10	0.9735
Stirmark_ROTSCALE_0.25	0.9473	Stirmark_RML_100	0.9812
Stirmark_ROTSCALE_-0.5	0.8968	Stirmark_SS_1	0.9870
Stirmark_ROT_0.25	0.9630	Stirmark_SS_2	1
Stirmark_ROT_-0.5	0.8913	Stirmark_SS_3	0.9564
Stirmark_ROTROP_-0.5	0.8957	Stirmark_ROTROP_0.25	0.9619

Watermark size 32 × 32, at $\Delta = 12$.			
Attacks	NC	Attacks	NC
Cropping 75% V	0.9969	Low pass 3×3	1
Cropping 50% V	0.9974	Wiener 3×3	1
Cropping 75% H	1	Median 3×3	1
Cropping 75% V+ H	0.9961	JPEG 75	1
Scale 0.75	1	JPEG 50	1
Scale 0.5	0.9900	JPEG 40	1
Gauss. noise m=0,v=0.002	0.9755	JPEG 30	1
S&P noise, $d=0.02$ + Median 3×3	1	S&P noise, $d=0.01$	0.9956
Contrast enhancements intensity=0.3, 0.9	1	Scale 2	1
Stirmark_AFFINE_1	0.9967	Stirmark_CONV_1	1
Stirmark_AFFINE_8	0.9760	Stirmark_RML_10	1
Stirmark_ROTSCALE_0.25	0.9923	Stirmark_RML_100	1
Stirmark_ROTSCALE_-0.5	0.9628	Stirmark_SS_1	1
Stirmark_ROT_0.25	0.9956	Stirmark_SS_2	1
Stirmark_ROT_-0.5	0.9749	Stirmark_SS_3	0.9934
Stirmark_ROTROP_-0.5	0.9727	Stirmark_ROTROP_0.25	0.9978

Table 4-5 Watermarked images after attacks

		
Cropping 75% V	Low pass 3×3	JPEG 75
		
Cropping 50% V	Wiener 3×3	JPEG 50
		
Cropping 75% H	Median 3×3	JPEG 25
		
Cropping 75% V+ H	S&P noise, $d=0.01$	Gaussian noise $m=0$, $v=0.002$
		
Scale 2	Scale 0.5	Gaussian noise $m=0$, $v=0.001$
		
S&P noise, $d=0.05$ + Median 3×3	Contrast enhancements intensity=0.3, 0.9	Contrast enhancements intensity=0.1, 0.5




















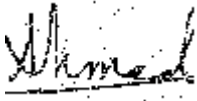
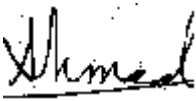

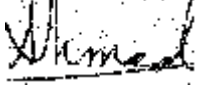




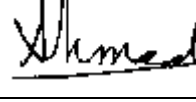
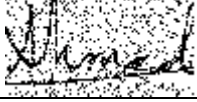
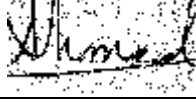
		
Stirmark_AFFINE_1	Stirmark_AFFINE_8	Stirmark_CONV_1
		
Stirmark_ROTSCALE_-0.5	Stirmark_ROTSCALE_0.25	Stirmark_ROT_0.25
		
Stirmark_ROT_-0.5	Stirmark_ROTSCALE_-0.5	Stirmark_ROTSCALE_0.25
		
Stirmark_SS_1	Stirmark_SS_2	Stirmark_SS_3
		
Stirmark_RML_10	Stirmark_RML_50	Stirmark_RML_100

Table 4-6 Reconstructed watermarks after attacks at $\Delta = 12$

		
Cropping 75% V	Low pass 3x3	JPEG 75
		
Cropping 50% V	Wiener 3x3	JPEG 40
		
Cropping 75% H	Median 3x3	JPEG 30
		
Cropping 75% V+H	S&P noise, $d=0.01$	Gaussian noise $m=0, v=0.002$
		
Scale 2	Scale 0.5	Gaussian noise $m=0, v=0.001$

S&P noise, $d=0.05+$ Median 3×3	Contrast enhancements intensity=0.3, 0.9	Contrast enhancements intensity=0.1, 0.5
Stirmark_AFFINE_1	Stirmark_AFFINE_8	Stirmark_CONV_1
Stirmark_ROTSCALE_-0.5	Stirmark_ROTSCALE_0.25	Stirmark_ROT_0.25
Stirmark_ROT_-0.5	Stirmark_ROTROP_-0.5	Stirmark_ROTROP_0.25
Stirmark_SS_1	Stirmark_SS_2	Stirmark_SS_3
Stirmark_RML_10	Stirmark_RML_50	Stirmark_RML_100

Table 4-7 Reconstructed watermarks after attacks at $\Delta=16$




Cropping 75% V	Low pass 3×3	JPEG 75
Cropping 50% V	Wiener 3×3	JPEG 30
Cropping 75% H	Median 3×3	JPEG 25
Cropping 75% V+H	S&P noise, $d=0.01$	Gaussian noise $m=0$, $v=0.002$
Scale 2	Scale 0.4	Gaussian noise $m=0$, $v=0.001$

S&P noise, $d=0.05+$ Median 3×3	Contrast enhancements intensity=0.3, 0.9	Contrast enhancements intensity=0.1, 0.5
Stirmark_AFFINE_1	Stirmark_AFFINE_8	Stirmark_CONV_1
Stirmark_ROTSCALE_-0.5	Stirmark_ROTSCALE_0.25	Stirmark_ROT_0.25
Stirmark_ROT_-0.5	Stirmark_ROTSCROP_-0.5	Stirmark_ROTSCROP_0.25
Stirmark_SS_1	Stirmark_SS_2	Stirmark_SS_3
Stirmark_RML_10	Stirmark_RML_50	Stirmark_RML_100

Table 4-8 Matlab execution time

Intel processor centrino 2 GHZ, RAM= 1 GB			
<i>Watermark size</i>	<i>Embedding time</i>	<i>Extraction time</i>	<i>Total time</i>
96×64	3.21 seconds	1.37 seconds	4.58 seconds
64×64	3.18 seconds	1.29 seconds	4.47 seconds
32×32	3.26 seconds	2.76 seconds	6.02 seconds

Table 4-9 Performance evaluation against high resolution images

Watermark size 96×64 at $\Delta = 12$.			
			
(a): Original Host image			
			
(b): Watermarked host image of size 1024×1024		(c): Watermarked host image of size 2048×2048	
PSNR	SSIM	PSNR	SSIM
39.9587	0.9709	40.0209	0.9695
Attacks	host image size 1024×1024	host image size 2048×2048	
JPEG 30	NC=0.9993	NC=1.0000	
Cropping 75 % both sides	NC=0.9808	NC=0.9611	
Low-pass Filter 3×3	NC=0.9991	NC=1.0000	
Median filter 3×3	NC=1.0000	NC=1.0000	
Wiener filter 3×3	NC=1.0000	NC=1.0000	

4.3 Algorithm 5: A Novel Blind Image Watermarking Technique for Colour RGB Images in the DCT Domain Using Green Channel

The presented algorithm in this section is also the colour version of algorithm 2 which was discussed in chapter 3. The aim of implementing algorithm 5 is to enhance the quality and robustness of algorithm 4 by using a different colour channel. The 24 bits/pixel RGB image is used and the watermark is placed on the green channel of the RGB image. The green channel was chosen after an analytical investigation process was carried out using some popular measurement metrics. The analysis and embedding processes have been carried out using DCT. The proposed algorithm has been shown to be resistant to JPEG compression, cropping, scaling, low-pass, median, removal attack and Stirmark attacks.

4.3.1 Analysis of Colour Images

In order to justify the reason for selecting the green channel for watermarking embedding over the red and blue channels some popular measurement metrics have been applied between the grey-scale version of the coloured image and each of the R,G,B colour components individually. A library of approximately 35 colour images and their associated grey level versions were used; each image is 24 bits/pixel RGB of size 512×512.

Analysis of the colour images have been carried out using the mean square error MSE and PSNR. They are used to compare the grey-scale image with RGB components. Thus, two images that are exactly the same will produce an infinite PSNR value and a zero MSE value. The DC values of the red, green and blue components were compared to the grey-scale version of the original image for a variety of images.

$$DC_Value = \frac{1}{\sqrt{XY}} \sum_{x,y} P_{x,y} \quad (4.1)$$

$$DC_Error = \frac{1}{\sqrt{XY}} \left[\sum_{x,y} P_{x,y} - P'_{x,y} \right] \quad (4.2)$$

Where p_{xy} is the grey-scale image and p'_{xy} is the individual R,G,B component respectively.

The results of the comparison are shown in Figures 4.3(a), 4.3(b),4.3(c) and 4.3(d). It is clear that the green channel is the closest to the grey image. In order to confirm the previous analytical results a fifth evaluation – as shown in Figure 4.3(e) – is carried out by calculating the SSIM percentage between the grey scale version of the host image and each of the R,G and B components. Since it was shown in algorithm 4 that embedding the watermark in the Y component (grey-scale information) of the YCrCb format of a colour image produces excellent invisible watermarking results, it becomes logical to choose the G channel of the RGB format for the embedding of the watermark. The reason for using the YCrCb format is that the human eye is less sensitive to chrominance than luminance. Compression algorithms can take advantage of this characteristic and subsample the values of Cb and Cr without significant visual degradation of the original color signal. The relation between YCrCb space and RGB space is given by the following linear transformation

$$\begin{aligned}
 Y &= 0.299R + 0.587G + 0.114B \\
 C_b &= 0.564(B - Y) + 128 \\
 C_r &= 0.713(R - Y) + 128
 \end{aligned}
 \tag{4.3}$$

Some researchers such as [35] have chosen to embed the watermark in each of the three RGB colour components individually and have compared the results for each colour component. It was concluded that the green channel is more robust to some attacks such as Gaussian noise while the blue channel is more robust for other attacks such as salt and pepper noise [35]. No invisibility qualities for the proposed method were mentioned. The main motivation for the proposed method is to show that embedding the watermark in the G channel of a colour image could produce better invisibility properties and robustness against several attacks compared to embedding in the blue channel.

4.3.2 The Embedding and Extraction Algorithms

In the technique presented here, the colour image is decomposed into three components R, G and B. Watermark information is embedded in the G plane to produce G' after embedding. The binary watermark digits are randomly scrambled using a secret key; this scrambling process is essential to reduce the spatial correlation between the host image and the embedded watermark. After the scrambling process, a shuffle scheme is applied for each binary watermark copy before embedding. Inside each 8×8 sub-blocks, 8 DCT coefficients are identified using the DCS; starting from these components the colour watermarked image will be recomposed. All the previous watermarking steps are described graphically in Figure 4.4. It is important to note that the watermark is embedded several times in the host image depending on the sizes of the host image and the watermark image. The use of multiple embedding, allows multiple watermarks to be inserted in an image with each watermark being independently verifiable.

The embedded watermarks information can be extracted by performing 8×8 DCT transform on the G channel of the watermarked host image and then indicating the same coefficients of the host image that carries the 8 bits of the embedded watermarks. A reverse shuffling scheme is implemented for the reconstructed watermarks. It is worth mentioning that although the proposed scheme is blind, it requires information such as the sizes of both the host and watermark images. By using the same secret key in the initial scrambling operation, the scrambled watermarks are descrambled to get the original watermarks. The extraction process is performed without needing the original unmarked image. Simply the recovery process is the inverse of the embedding process as shown in Figure 4.5. The following information must be present for the extraction process: the size of the host image, the size of the watermark image and the watermark embedding strength Δ .

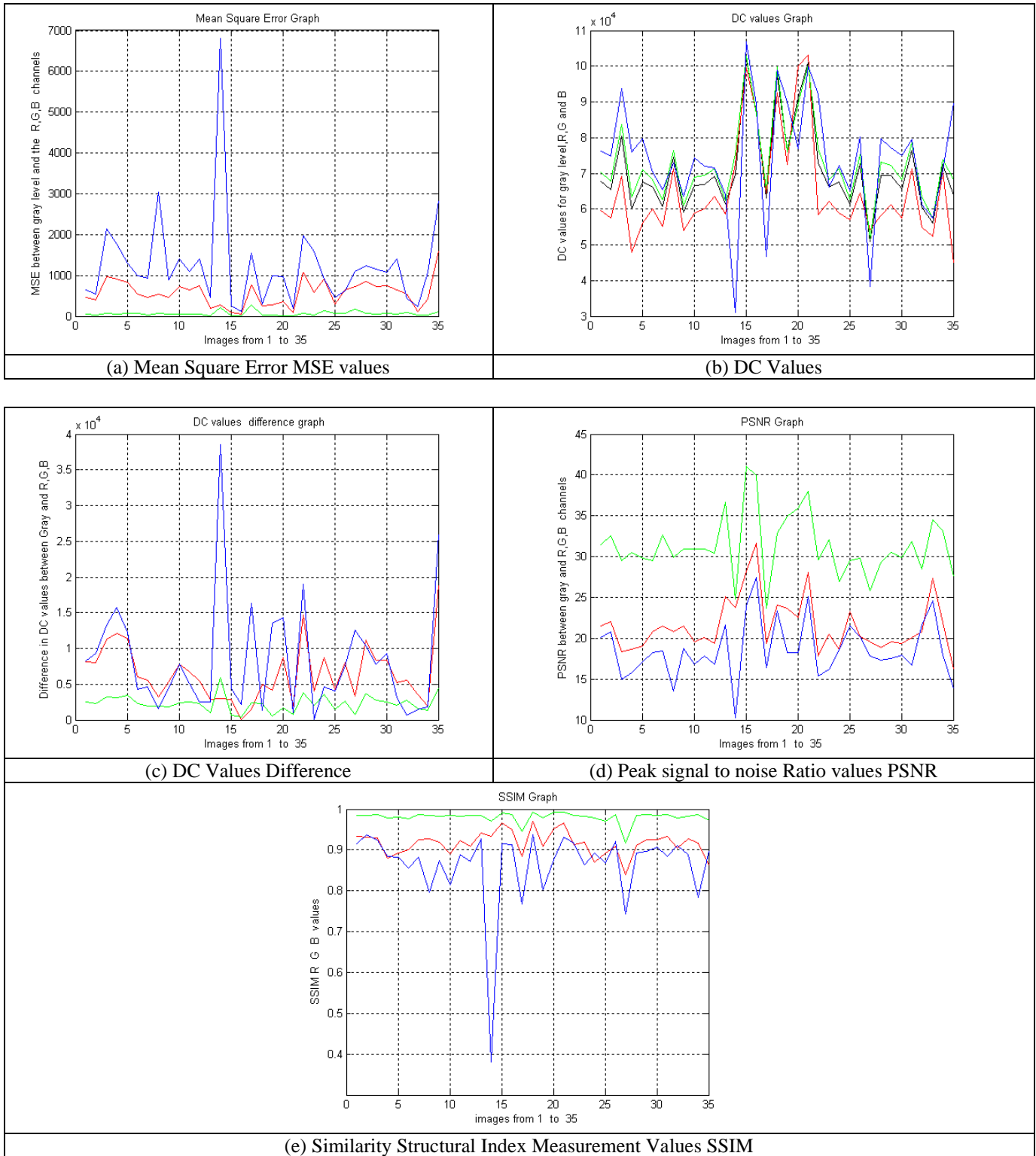


Figure 4-3 Analysis between grey-scale images and each of R,G,B components

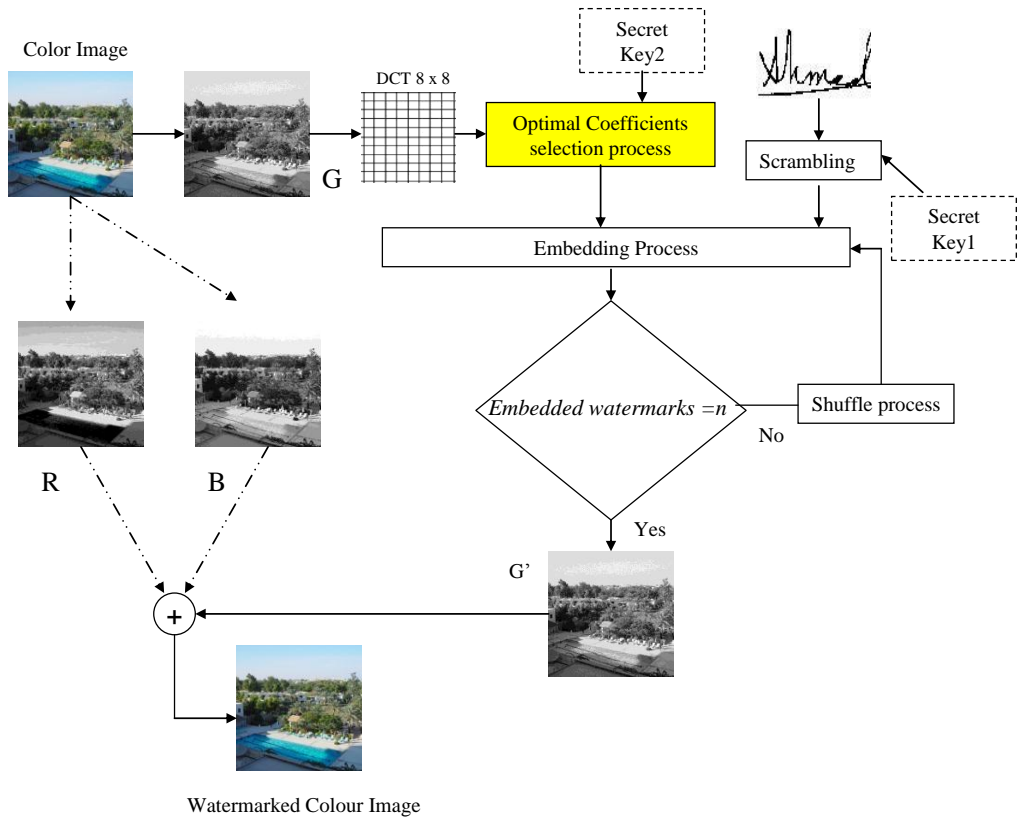


Figure 4-4 Graphical presentation for embedding steps

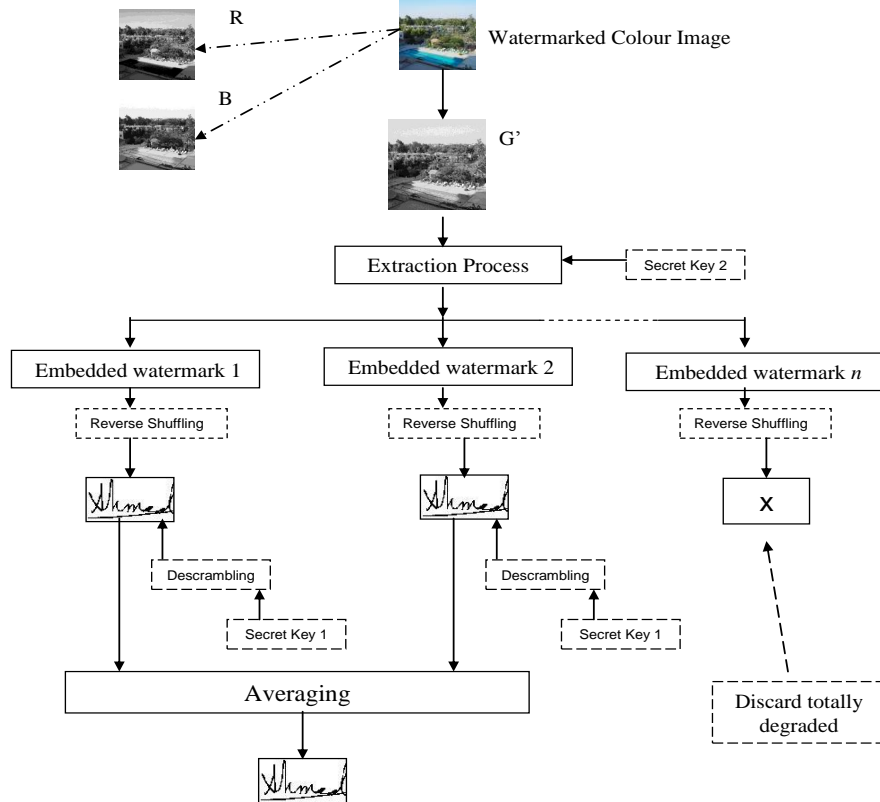


Figure 4-5 Graphical presentation for extraction steps

4.3.3 Simulation and Results

This algorithm is examined using different colour images of size 512×512 with 24 bits per pixel. Also 3 different hand written signatures of sizes 96×64 , 64×64 and 32×32 are used as watermarks. The perceptual invisibility is evaluated using PSNR at different embedding strengths and different watermark sizes as shown in Table 4.10. The average PSNR values between the watermarked and original images using a watermark size of 32×32 are 49 dB and 38.6 dB for watermarking strengths $\Delta = 8$ and $\Delta = 34$, respectively. If watermarks of size 64×64 are used, then the average PSNR values will be 48.9 dB and 38.4 dB for $\Delta = 8$ and $\Delta = 34$, respectively. Finally, the average PSNR values using a watermark size of 96×64 are 48.8 dB and 38.2 dB for $\Delta = 8$ and $\Delta = 34$, respectively. In Table 4.11 the perceptual invisibility of the proposed algorithm is evaluated using SSIM at different embedding strengths and different watermark sizes. The original “Lena” colour image has been used to examine the perceptual quality at different embedding strengths as depicted in Table 4.12.

To verify the robustness of the proposed method, various common signal processing and geometric attacks are applied to the watermarked images. NC is used to measure the similarity between the original and the extracted watermark as shown in Table 4.13.

Various embedding strengths have been investigated to determine which values provide the best performance for the majority of the images. Visual inspection of the extracted watermarks using different embedding strengths at $\Delta = 24$ and $\Delta = 34$ are depicted in Tables 4.15 and 4.16 respectively. The experimental results show that the performance achieved by the proposed method for the extracted watermark after running different attacks is perceptually visible at $\Delta = 24$, which can be considered as the best value for the embedding strength. A higher embedding strength value such as $\Delta = 34$ will provide strong robustness and distinct perceptual visibility for the extracted watermark. It is worth noting that higher embedding strength could reduce the invisibility qualities as demonstrated in Table 4.10. The Matlab execution time on a 2Ghz Centrino processor and 1Gb memory is shown in Table 4.17. Finally, the performance evaluation against high resolution images is illustrated in Table 4.18.

Table 4-10 PSNR for watermarked colour images

Watermark size 96 × 64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
PSNR at $\Delta = 8$	48.8089 dB	48.3071 dB	49.1091 dB
PSNR at $\Delta = 12$	45.8017 dB	45.6790 dB	46.1119 dB
PSNR at $\Delta = 24$	40.8101 dB	40.4071 dB	40.9901 dB
PSNR at $\Delta = 30$	39.1001 dB	38.8124 dB	39.7481 dB
PSNR at $\Delta = 34$	38.2083 dB	38.0970 dB	38.8093 dB
Watermark size 64 × 64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
PSNR at $\Delta = 8$	48.9081 dB	48.5072 dB	49.1091 dB
PSNR at $\Delta = 12$	45.9079 dB	45.8745 dB	46.1119 dB
PSNR at $\Delta = 24$	40.9178 dB	40.7061 dB	40.9901 dB
PSNR at $\Delta = 30$	39.4131 dB	38.9904 dB	39.7481 dB
PSNR at $\Delta = 34$	38.4033 dB	38.3140 dB	38.8093 dB
Watermark size 32 × 32			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
PSNR at $\Delta = 8$	49.0023 dB	48.7056 dB	49.3091 dB
PSNR at $\Delta = 12$	46.0045 dB	45.9985 dB	46.4156 dB
PSNR at $\Delta = 24$	41.0165 dB	40.9803 dB	41.0513 dB
PSNR at $\Delta = 30$	39.7120 dB	39.0911 dB	39.9361 dB
PSNR at $\Delta = 34$	38.6055 dB	38.5651 dB	38.9123 dB

Table 4-11 SSIM for watermarked colour images

Watermark size 96 × 64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
SSIM at $\Delta = 8$	0.9949	0.9944	0.9982
SSIM at $\Delta = 12$	0.9904	0.9871	0.9954
SSIM at $\Delta = 24$	0.9705	0.9688	0.9880
SSIM at $\Delta = 30$	0.9681	0.9522	0.9826
SSIM at $\Delta = 34$	0.9511	0.9398	0.9785
Watermark size 64 × 64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
SSIM at $\Delta = 8$	0.9953	0.9945	0.9985
SSIM at $\Delta = 12$	0.9912	0.9899	0.9961
SSIM at $\Delta = 24$	0.9762	0.9714	0.9885
SSIM at $\Delta = 30$	0.9674	0.9559	0.9838
SSIM at $\Delta = 34$	0.9583	0.9415	0.9797
Watermark size 32 × 32			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
SSIM at $\Delta = 8$	0.9959	0.9946	0.9987
SSIM at $\Delta = 12$	0.9919	0.9911	0.9964
SSIM at $\Delta = 24$	0.9769	0.9742	0.9887
SSIM at $\Delta = 30$	0.9685	0.9615	0.9839
SSIM at $\Delta = 34$	0.9631	0.9540	0.9807

Table 4-12 Original and watermarked Lena images at different embedding strengths

























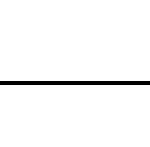
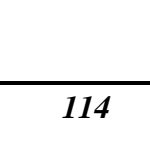
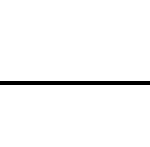
	
Original un-watermarked Lena image	
	
Watermarked image at $\Delta = 12$	Watermarked image at $\Delta = 24$
	
Watermarked image at $\Delta = 30$	Watermarked image at $\Delta = 34$

Table 4-13 Normalized correlation for Lena colour image

Watermark size 96×64, at $\Delta = 24$.			
Attacks	NC	Attacks	NC
Cropping 75% V	0.9721	Low pass 3×3	0.9901
Cropping 50% V	0.9891	Low pass 5×5	0.9541
Cropping 75% H	1	Wiener 3×3	0.9912
Cropping 75% V+ H	0.9843	Wiener 5×5	0.9734
Scale 2	1	Median 3×3	0.9972
Scale 0.75	0.9910	Median 5×5	0.9810
Gaussian noise m=0, v=0.002	0.8545	JPEG 75	0.9995
Gaussian noise m=0, v=0.001	0.9861	JPEG 50	0.9975
S&P noise, d=0.02+ Median 3×3	0.9912	JPEG 25	0.9845
S&P noise, d=0.05+ Median 3×3	0.9903	JPEG 20	0.9762
Contrast enhancements intensity=0.3, 0.9	0.9963	Scale 0.4	0.8884
Contrast enhancements intensity=0.1, 0.5	0.9512	S&P noise, d=0.02	0.7912
Stirmark_AFFINE_1	0.9454	Stirmark_CONV_1	0.9573
Stirmark_AFFINE_8	0.8856	Stirmark_RML_10	0.9819
Stirmark_ROTSCALE_0.25	0.9401	Stirmark_RML_100	0.9960
Stirmark_ROTSCALE_-0.5	0.8724	Stirmark_SS_1	0.9996
Stirmark_ROT_0.25	0.9442	Stirmark_SS_2	1
Stirmark_ROT_-0.5	0.8651	Stirmark_SS_3	0.9892
Stirmark_ROTROP_-0.5	0.8681	Stirmark_ROTROP_0.25	0.9495
Watermark size 64×64, at $\Delta = 24$.			
Attacks	NC	Attacks	NC
Cropping 75% V	0.9832	Low pass 3×3	0.9923
Cropping 50% V	0.9923	Low pass 5×5	0.9712
Cropping 75% H	1	Wiener 3×3	1
Cropping 75% V+ H	0.9856	Wiener 5×5	0.9865
Scale 2	1	Median 3×3	1
Scale 0.75	0.9954	Median 5×5	0.9976
Gaussian noise m=0, v=0.002	0.9612	JPEG 75	1
Gaussian noise m=0, v=0.001	0.9965	JPEG 50	1
S&P noise, d=0.02+ Median 3×3	0.9978	JPEG 25	0.9919
S&P noise, d=0.05+ Median 3×3	0.9934	JPEG 20	0.9801
Contrast enhancements intensity=0.3, 0.9	0.9945	Scale 0.4	0.9430
Contrast enhancements intensity=0.1, 0.5	0.9712	S&P noise, d=0.02	0.9112
Stirmark_AFFINE_1	0.9788	Stirmark_CONV_1	0.9752
Stirmark_AFFINE_8	0.9487	Stirmark_RML_10	0.9898
Stirmark_ROTSCALE_0.25	0.9752	Stirmark_RML_100	0.9876
Stirmark_ROTSCALE_-0.5	0.9343	Stirmark_SS_1	0.9997
Stirmark_ROT_0.25	0.9823	Stirmark_SS_2	1
Stirmark_ROT_-0.5	0.9398	Stirmark_SS_3	0.9917
Stirmark_ROTROP_-0.5	0.9437	Stirmark_ROTROP_0.25	0.9788
Watermark size 32×32, at $\Delta = 24$.			
Attacks	NC	Attacks	NC
Cropping 75% V	0.9901	Low pass 3×3	1
Cropping 50% V	0.9986	Low pass 5×5	1
Cropping 75% H	1	Wiener 3×3	1
Cropping 75% V+ H	0.9901	Wiener 5×5	1
Scale 2	1	Median 3×3	1
Scale 0.75	1	Median 5×5	1
Gaussian noise m=0, v=0.002	0.9976	JPEG 75	1
Gaussian noise m=0, v=0.001	1	JPEG 50	1
S&P noise, d=0.02+ Median 3×3	0.9987	JPEG 25	1
S&P noise, d=0.05+ Median 3×3	0.9954	JPEG 20	1
Contrast enhancements intensity=0.3, 0.9	1	Scale 0.4	0.9941
Contrast enhancements intensity=0.1, 0.5	0.9944	S&P noise, d=0.02	0.9756
Stirmark_AFFINE_1	0.9978	Stirmark_CONV_1	1
Stirmark_AFFINE_8	0.9923	Stirmark_RML_10	1

Stirmark_ROTSCALE_0.25	0.9956	Stirmark_RML_100	1
Stirmark_ROTSCALE_-0.5	0.9913	Stirmark_SS_1	1
Stirmark_ROT_0.25	1	Stirmark_SS_2	1
Stirmark_ROT_-0.5	0.9847	Stirmark_SS_3	1
Stirmark_ROTFCROP_-0.5	0.9913	Stirmark_ROTFCROP_0.25	1

Table 4-14 Watermarked images after attacks

		
Cropping 75% V	Low pass 3×3	JPEG 75
		
Cropping 50% V	Wiener 3×3	JPEG 25
		
Cropping 75% H	Median 3×3	JPEG 20
		
Cropping 75% V+ H	S&P noise, $d=0.01$	Gaussian noise $m=0, v=0.002$
		
Scale 2	Scale 0.5	Gaussian noise $m=0, v=0.001$
		
S&P noise, $d=0.05$ + Median 3×3	Contrast enhancements intensity=0.3, 0.9	Contrast enhancements intensity=0.1, 0.5
		
Stirmark_AFFINE_1	Stirmark_AFFINE_8	Stirmark_CONV_1
		
Stirmark_ROTSCALE_-0.5	Stirmark_ROTSCALE_0.25	Stirmark_ROT_0.25
		












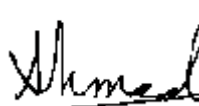
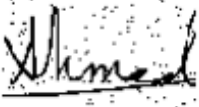
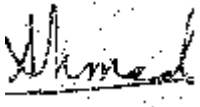


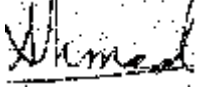




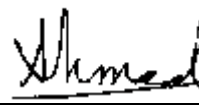

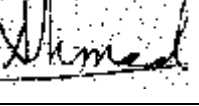
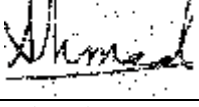
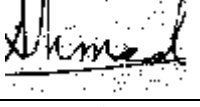
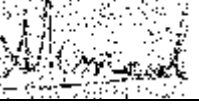
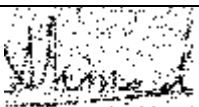





		
Stirmark_ROT_-0.5	Stirmark_ROTcrop_-0.5	Stirmark_ROTcrop_0.25
		
Stirmark_SS_1	Stirmark_SS_2	Stirmark_SS_3
		
Stirmark_RML_10	Stirmark_RML_50	Stirmark_RML_100

Table 4-15 Reconstructed watermarks after attacks at $\Delta = 24$

		
Cropping 75% V	Low pass 3x3	JPEG 75
		
Cropping 50% V	Wiener 3x3	JPEG 25
		
Cropping 75% H	Median 3x3	JPEG 20
		
Cropping 75% V+H	S&P noise, $d=0.01$	Gaussian noise $m=0, v=0.002$
		
Scale 2	Scale 0.4	Gaussian noise $m=0, v=0.001$
		
S&P noise, $d=0.05$ + Median 3x3	Contrast enhancements intensity=0.3, 0.9	Contrast enhancements intensity=0.1, 0.5
		
Stirmark_AFFINE_1	Stirmark_AFFINE_8	Stirmark_CONV_1
		

Stirmark_ROTSCALE_-0.5	Stirmark_ROTSCALE_0.25	Stirmark_ROT_0.25
Stirmark_ROT_-0.5	Stirmark_ROT_CROP_-0.5	Stirmark_ROT_CROP_0.25
Stirmark_SS_1	Stirmark_SS_2	Stirmark_SS_3
Stirmark_RML_10	Stirmark_RML_50	Stirmark_RML_100

Table 4-16 Reconstructed watermarks after attacks at $\Delta = 34$

Cropping 75% V	Low pass 3x3	JPEG 75
Cropping 50% V	Wiener 3x3	JPEG 20
Cropping 75% H	Median 3x3	JPEG 15
Cropping 75% V+H	S&P noise, $d=0.01$	Gaussian noise $m=0, v=0.002$
Scale 2	Scale 0.4	Gaussian noise $m=0, v=0.001$
S&P noise, $d=0.05+$ Median 3x3	Contrast enhancements intensity=0.3, 0.9	Contrast enhancements intensity=0.1, 0.5
Stirmark_AFFINE_1	Stirmark_AFFINE_8	Stirmark_CONV_1
Stirmark_ROTSCALE_-0.5	Stirmark_ROTSCALE_0.25	Stirmark_ROT_0.25



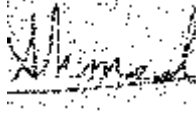



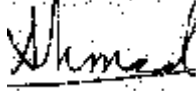
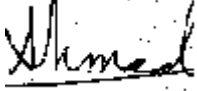

		
Stirring_ROT_-0.5	Stirring_ROT_CROP_-0.5	Stirring_ROT_CROP_0.25
		
Stirring_SS_1	Stirring_SS_2	Stirring_SS_3
		
Stirring_RML_10	Stirring_RML_50	Stirring_RML_100

Table 4-17 Matlab execution time

Intel processor centrino 2 GHZ, RAM= 1 GB			
<i>Watermark size</i>	<i>Embedding time</i>	<i>Extraction time</i>	<i>Total time</i>
96×64	3.25 seconds	1.34 seconds	4.59 seconds
64×64	3.17 seconds	1.43 seconds	4.60 seconds
32×32	3.15 seconds	3.17 seconds	6.32 seconds

Table 4-18 Performance evaluation against high resolution images

Watermark size 96×64 at $\Delta = 24$.



(a): Original Host image



(b): Watermarked host image of size 1024×1024

(c): Watermarked host image of size 2048×2048

PSNR	SSIM	PSNR	SSIM
40.5791	0.9780	40.6640	0.9769

Attacks	host image size 1024×1024	host image size 2048×2048
JPEG 30	NC=1.0000	NC=1.0000
Cropping 75 % both sides	NC=0.9843	NC=0.9679
Low-pass Filter 3×3	NC=1.0000	NC=1.0000
Median filter 3×3	NC=1.0000	NC=1.0000
Wiener filter 3×3	NC=1.0000	NC=1.0000

4.4 Algorithm 6: A High Capacity Watermarking Technique for the Copyright protection of Colour Images

This proposed technique here is the colour version of algorithm 3 of chapter 3. After investigating different colour channels in algorithm 4 and 5, the green channel is

selected as a good channel to accommodate the watermark over Y, R and B channels. The proposed algorithm can embed watermarking information using 25% of the host image size instead of 12.5% in algorithms 4 and 5. For example, if the watermark size is as big as the one used in this proposed technique (224×128), the previous watermarking algorithms (algorithms 4 and 5) that are based on embedding a single watermark multiple times in the host images will fail to repeat the watermark through the host image. The technique can resist classical attacks such as JPEG compression, low pass filtering, median filtering, cropping, and scaling attacks.

4.4.1 Embedding and Extraction Steps

In this approach, a block DCT-based algorithm is developed to embed the binary watermark into the green channel of the colour host image. The very low frequency components of the colour host image will be utilized during the watermark embedding. Sixteen bits are embedded inside each 8×8 DCT sub-blocks of the host image. The 16 frequency coefficients are predefined after a zigzag process. The zigzag process progresses from low-frequency to high-frequency terms where the first sixteen low frequencies excluding the DC coefficient will be selected. The proposed watermarking scheme is based on the possibility of embedding multiple copies of the binary watermark(s) in the host image (depending on the sizes of the host image and the watermark image). This will increase the robustness of the watermark against several attacks such as cropping and compression. The binary watermark digits are randomly scrambled using a secret key; after the scrambling process, a shuffle scheme is applied for each binary watermark copy before embedding. The number of watermark shifted bits depends on the sizes of the host image and the watermark.

The watermark information can be retrieved by indicating the same coefficients of the host image that carries the 16 bits of the embedded watermarks. The watermark must be descrambled and a reverse process to the shuffle scheme must be applied.

The totally degraded copy of the extracted watermarks must be discarded and then averaging process is applied, the resultant average watermark is selected as the final reconstructed watermark or one copy of the extracted watermarks is selected as the final watermark if it provides better results than the resultant average watermark. All the previous watermarking steps are described graphically in the diagram as shown in Figures 4.6 and 4.7.

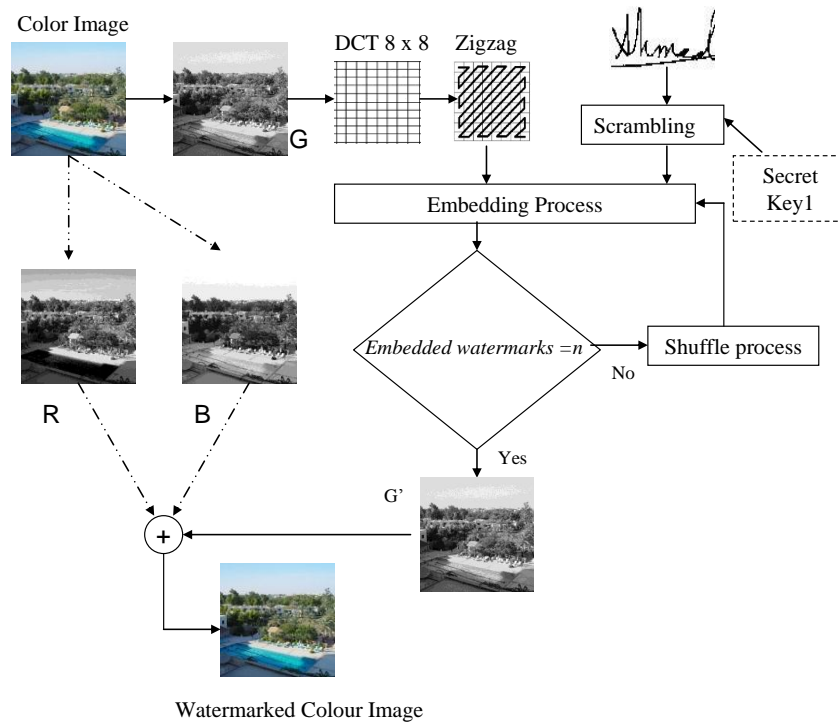


Figure 4-6 Graphical presentation for the embedding steps

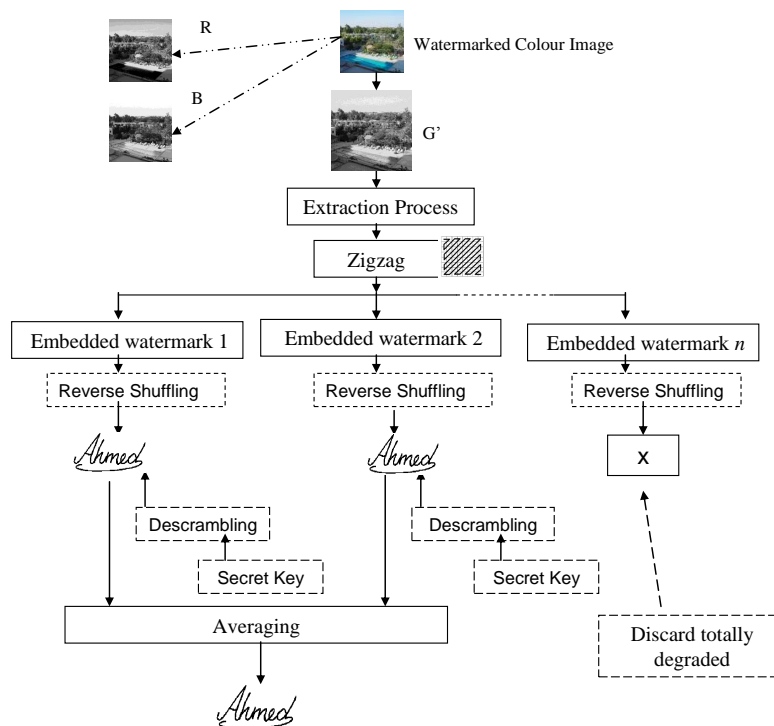


Figure 4-7 Graphical presentation for the extraction steps

4.4.2 Simulation and Results

This algorithm is examined using different colour images of size 512×512 with 24 bits per pixel. Also a hand written signature image of size 224×128 is used as watermark. Different watermarking strengths Δ have been investigated and the performance has been evaluated in the experiments. Table 4.19 demonstrates the perceptual invisibility of the proposed algorithm at different embedding strengths. The PSNR values between the watermarks and the original images when using a watermark size of size 224×128 are 45.7 dB and 35.8 dB for watermarking strengths $\Delta = 12$ and $\Delta = 34$, respectively. In Table 4.20 the perceptual invisibility of the proposed algorithm is evaluated using SSIM at different embedding strengths. The original “Lena” colour image has been used to examine the perceptual quality at different embedding strengths as depicted in Table 4.21.

Since the robustness and visibility of a digital watermark are directly related, an increase in the watermark size also increases the visibility of the watermark. Various embedding strengths have been investigated to determine which values provide the best performance for the majority of the images. Table 4.22 shows the effect of different watermark sizes in the proposed technique. It has been found that the best overall performance achieved by the proposed method, extracting the watermark

after running different attacks, is at $\Delta = 24$, when the watermarking becomes perceptually visible. Higher embedding strength values such as $\Delta = 34$ will provide stronger robustness, but will introduce more distortion in the host images.

To verify the robustness of the proposed method, various common signal processing and geometric attacks were applied to the watermarked images. NC is used to measure the similarity between the original and the extracted watermark as shown in Table 4.22. The smaller the number of pixels in the watermark image, the more the watermark can be repeated throughout the host image which in turn increases the robustness. A visual inspection of the extracted watermarks is depicted in Table 4.24 when $\Delta = 24$ and Table 4.25 when $\Delta = 34$. The execution time of algorithm 6 with different watermark sizes is shown in Table 4.26. Finally, the performance evaluation against high resolution images is illustrated in Table 4.27.

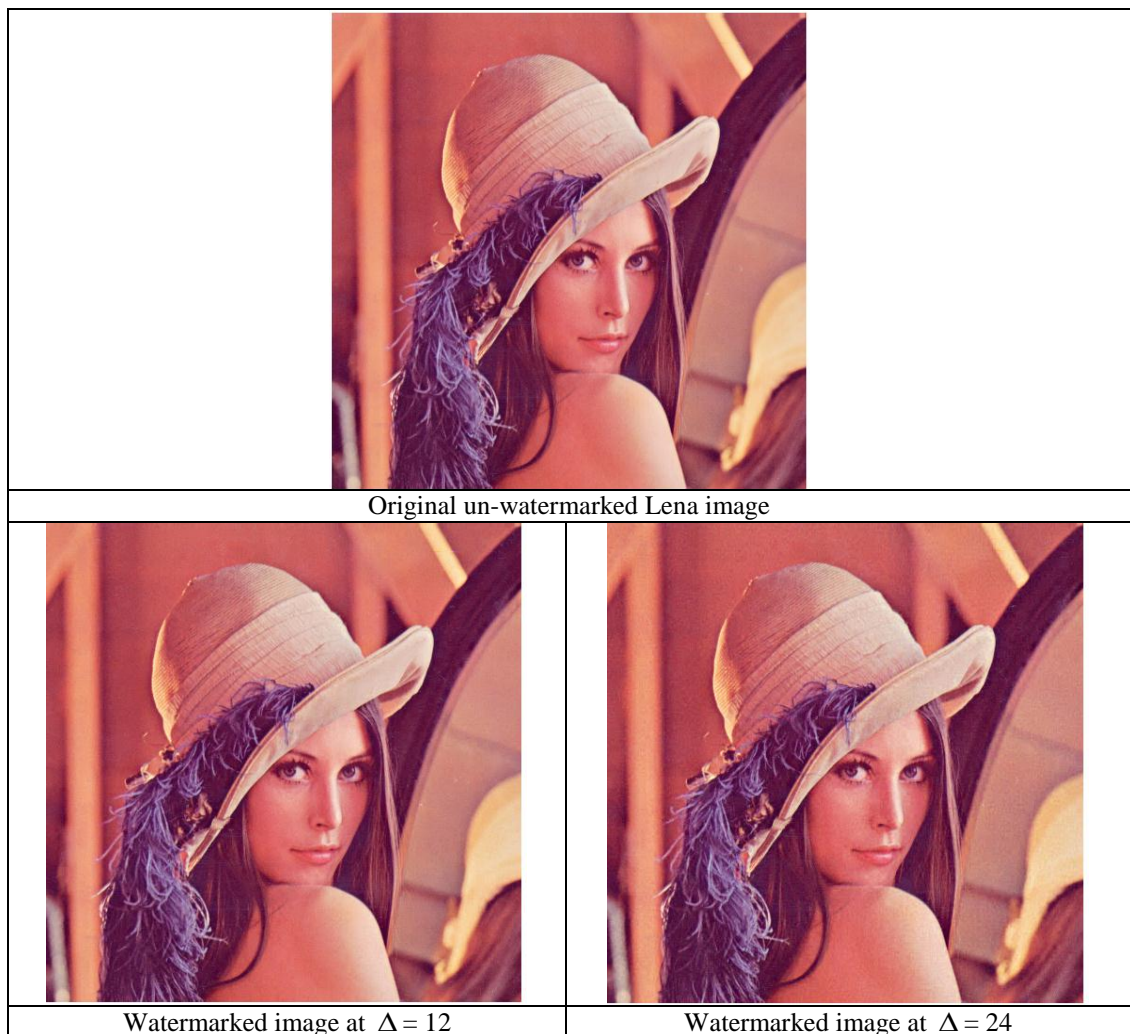
Table 4-19 PSNR for watermarked colour images

Watermark size 224 × 128			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
PSNR at $\Delta = 8$	45.7487 dB	45.1712 dB	46.3997 dB
PSNR at $\Delta = 12$	43.0611 dB	41.2531 dB	43.2511 dB
PSNR at $\Delta = 24$	38.5134 dB	35.8812 dB	37.6145 dB
PSNR at $\Delta = 30$	36.9573 dB	33.9482 dB	35.8926 dB
PSNR at $\Delta = 34$	35.8011 dB	33.0131 dB	34.8076 dB
Watermark size 224 × 96			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
PSNR at $\Delta = 8$	45.9481 dB	45.3011 dB	46.6997 dB
PSNR at $\Delta = 12$	43.2610 dB	41.3230 dB	43.4123 dB
PSNR at $\Delta = 24$	38.7120 dB	36.1830 dB	37.7012 dB
PSNR at $\Delta = 30$	37.1003 dB	34.1002 dB	35.9762 dB
PSNR at $\Delta = 34$	35.9101 dB	33.3001 dB	35.0123 dB
Watermark size 192 × 64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
PSNR at $\Delta = 8$	46.1081 dB	45.5111 dB	46.9007 dB
PSNR at $\Delta = 12$	43.3010 dB	41.4150 dB	43.6103 dB
PSNR at $\Delta = 24$	38.8001 dB	36.3410 dB	37.9012 dB
PSNR at $\Delta = 30$	37.2204 dB	34.3112 dB	36.1042 dB
PSNR at $\Delta = 34$	36.1001 dB	33.5312 dB	35.3033 dB

Table 4-20 SSIM for watermarked colour images

Watermark size 224 × 128			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
SSIM at $\Delta = 8$	0.9907	0.9894	0.9965
SSIM at $\Delta = 12$	0.9830	0.9796	0.9927
SSIM at $\Delta = 24$	0.9541	0.9408	0.9777
SSIM at $\Delta = 30$	0.9398	0.9221	0.9694
SSIM at $\Delta = 34$	0.9278	0.9063	0.9633
Watermark size 224 × 96			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
SSIM at $\Delta = 8$	0.9907	0.9898	0.9965
SSIM at $\Delta = 12$	0.9835	0.9812	0.9928
SSIM at $\Delta = 24$	0.9573	0.9465	0.9781
SSIM at $\Delta = 30$	0.9429	0.9278	0.9700
SSIM at $\Delta = 34$	0.9334	0.9146	0.9642
Watermark size 192 × 64			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
SSIM at $\Delta = 8$	0.9909	0.9907	0.9965
SSIM at $\Delta = 12$	0.9843	0.9833	0.9928
SSIM at $\Delta = 24$	0.9615	0.9547	0.9789
SSIM at $\Delta = 30$	0.9489	0.9385	0.9707
SSIM at $\Delta = 34$	0.9399	0.9273	0.9658

Table 4-21 Original and watermarked Lena images at different embedding strengths



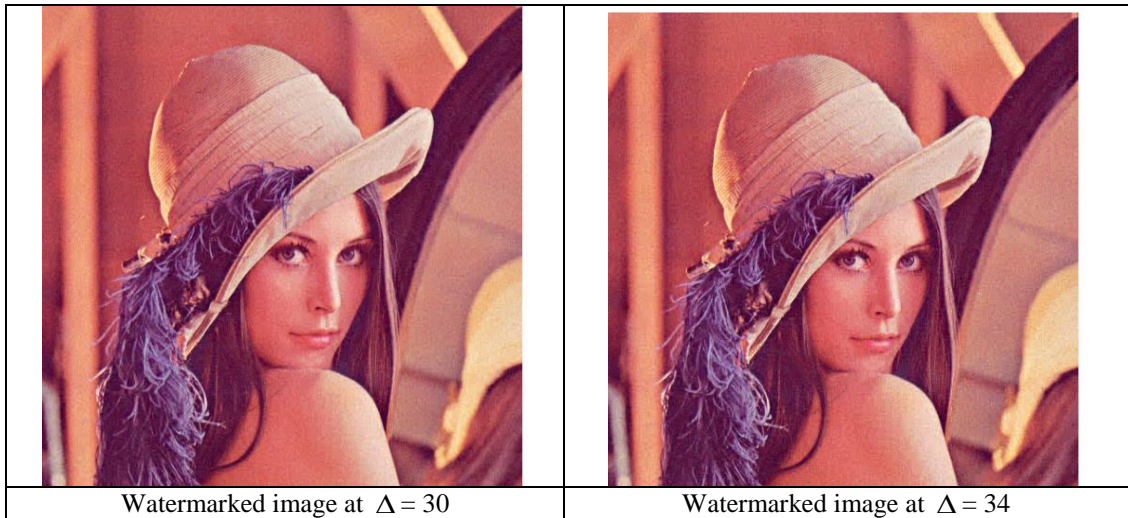
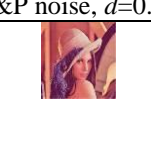


Table 4-22 Normalized correlation for the Lena colour image

Watermark size 224×128 , at $\Delta = 24$.			
Attacks	NC	Attacks	NC
Cropping 75% V	0.9753	Low pass 3×3	0.9935
Cropping 50% V	0.9800	Wiener 3×3	0.9989
Cropping 75% H	0.9909	Median 3×3	0.9965
Cropping 75% V+H	0.9919	JPEG 75	0.9988
Gauss. noise $m=0, v=0.002$	0.9161	JPEG 50	0.9953
Gaussian noise $m=0, v=0.001$	0.9384	JPEG 40	0.9950
S&P noise, $d=0.02$ + Median 3×3	0.9872	JPEG 25	0.9946
S&P noise, $d=0.05$ + Median 3×3	0.9944	JPEG 20	0.9885
Contrast enhancements intensity=0.3, 0.9	0.9920	S&P noise, $d=0.02$	0.8776
Contrast enhancements intensity=0.1, 0.5	0.9829	Scale 2	1
Scale 0.5	0.9428	Scale 0.75	0.9864
Stirmark_AFFINE_1	0.9690	Stirmark_CONV_1	0.9822
Stirmark_AFFINE_8	0.9480	Stirmark_RML_10	0.9812
Stirmark_ROTSCALE_0.25	0.9655	Stirmark_RML_100	0.9935
Stirmark_ROTSCALE_-0.5	0.9429	Stirmark_SS_1	0.9998
Stirmark_ROT_0.25	0.9684	Stirmark_SS_2	1
Stirmark_ROT_-0.5	0.9435	Stirmark_SS_3	0.9932
Stirmark_ROTROP_-0.5	0.9456	Stirmark_ROTROP_0.25	0.9680
Watermark size 224×96 , at $\Delta = 24$.			
Attacks	NC	Attacks	NC
Cropping 75% V	0.9752	Low pass 3×3	0.9702
Cropping 50% V	0.9798	Wiener 3×3	0.9932
Cropping 75% H	0.9908	Median 3×3	0.9793
Cropping 75% V+H	0.9917	JPEG 75	0.9931
Gaussian noise $m=0, v=0.002$	0.7984	JPEG 50	0.9822
Gaussian noise $m=0, v=0.001$	0.9482	JPEG 40	0.9801
S&P noise, $d=0.02$ + Median 3×3	0.9772	JPEG 25	0.9742
S&P noise, $d=0.05$ + Median 3×3	0.9713	JPEG 20	0.9631
Contrast enhancements intensity=0.3, 0.9	0.9765	S&P noise, $d=0.02$	0.7333
Contrast enhancements intensity=0.1, 0.5	0.9414	Scale 2	1
Scale 0.5	0.8780	Scale 0.75	0.9462
Stirmark_AFFINE_1	0.9240	Stirmark_CONV_1	0.9440
Stirmark_AFFINE_8	0.8669	Stirmark_RML_10	0.9400
Stirmark_ROTSCALE_0.25	0.9057	Stirmark_RML_100	0.9769
Stirmark_ROTSCALE_-0.5	0.8534	Stirmark_SS_1	0.9957
Stirmark_ROT_0.25	0.9068	Stirmark_SS_2	1
Stirmark_ROT_-0.5	0.8544	Stirmark_SS_3	0.9646
Stirmark_ROTROP_-0.5	0.8521	Stirmark_ROTROP_0.25	0.9180
Watermark size 192×64 , at $\Delta = 24$.			

Attacks	NC	Attacks	NC
Cropping 75% V	0.9738	Low pass 3×3	0.9934
Cropping 50% V	0.9801	Wiener 3×3	0.9995
Cropping 75% H	0.9912	Median 3×3	0.9980
Cropping 75% V+ H	0.9921	JPEG 75	0.9995
Gaussian noise m=0, v=0.002	0.8542	JPEG 50	0.9960
Gaussian noise m=0, v=0.001	0.9782	JPEG 40	0.9950
S&P noise, $d=0.02$ + Median 3×3	0.9976	JPEG 25	0.9934
S&P noise, $d=0.05$ + Median 3×3	0.9953	JPEG 20	0.9829
Contrast enhancements intensity=0.3, 0.9	0.9665	S&P noise, $d=0.02$	0.7751
Contrast enhancements intensity=0.1, 0.5	0.9314	Scale 2	1
Scale 0.5	0.9324	Scale 0.75	0.9842
Stirmark_AFFINE_1	0.9623	Stirmark_CONV_1	0.9768
Stirmark_AFFINE_8	0.9243	Stirmark_RML_10	0.9806
Stirmark_ROTSCALE_0.25	0.9591	Stirmark_RML_100	0.9968
Stirmark_ROTSCALE_-0.5	0.9142	Stirmark_SS_1	0.9998
Stirmark_ROT_0.25	0.9610	Stirmark_SS_2	1
Stirmark_ROT_-0.5	0.9177	Stirmark_SS_3	0.9908
Stirmark_ROTSCROP_-0.5	0.9158	Stirmark_ROTSCROP_0.25	0.9622

Table 4-23 Watermarked images after attacks

		
Cropping 75% V	Low pass 3×3	JPEG 75
		
Cropping 50% V	Wiener 3×3	JPEG 25
		
Cropping 75% H	Median 3×3	JPEG 20
		
Cropping 75% V+ H	S&P noise, $d=0.01$	Gaussian noise m=0, v=0.002
		
Scale 2	Scale 0.5	Gaussian noise m=0, v=0.001
		
S&P noise, $d=0.05$ + Median 3×3	Contrast enhancements intensity=0.3, 0.9	Contrast enhancements intensity=0.1, 0.5



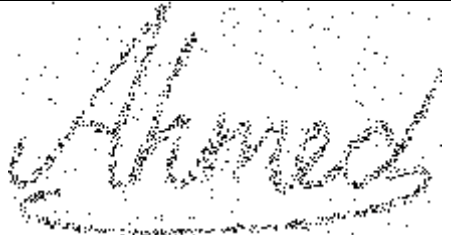


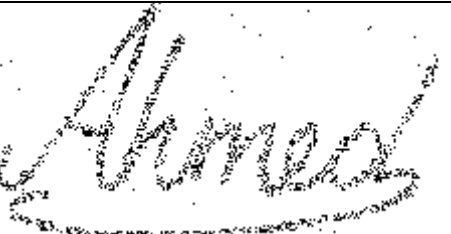
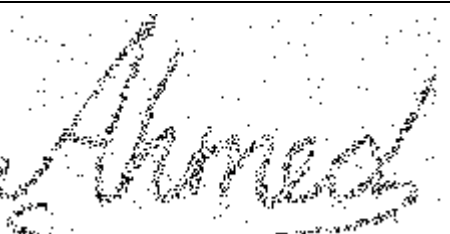
		
Stirmark_AFFINE_1	Stirmark_AFFINE_8	Stirmark_CONV_1
		
Stirmark_ROTSCALE_-0.5	Stirmark_ROTSCALE_0.25	Stirmark_ROT_0.25
		
Stirmark_ROT_-0.5	Stirmark_ROT_CROP_-0.5	Stirmark_ROT_CROP_0.25
		
Stirmark_SS_1	Stirmark_SS_2	Stirmark_SS_3
		
Stirmark_RML_10	Stirmark_RML_50	Stirmark_RML_100

Table 4-24 Reconstructed watermarks after attacks at $\Delta = 24$

		
Cropping 75% V	Low pass 3x3	JPEG 75
		
Cropping 50% V	Wiener 3x3	JPEG 25

Cropping 75% H	Median 3×3	JPEG 20
Cropping 75% V+H	S&P noise, $d=0.02$	Gaussian noise $m=0, v=0.002$
Scale 2	Scale 0.5	Gaussian noise $m=0, v=0.001$
S&P noise, $d=0.05+$ Median 3×3	Contrast enhancements intensity=0.3, 0.9	Contrast enhancements intensity=0.1, 0.5
Stirmark_AFFINE_1	Stirmark_AFFINE_8	Stirmark_CONV_1
Stirmark_ROTSCALE_-0.5	Stirmark_ROTSCALE_0.25	Stirmark_ROT_0.25


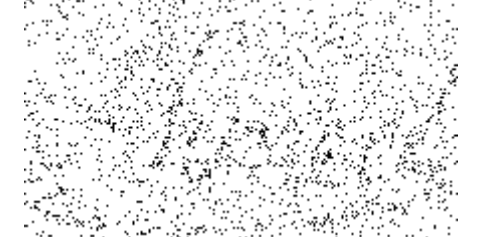
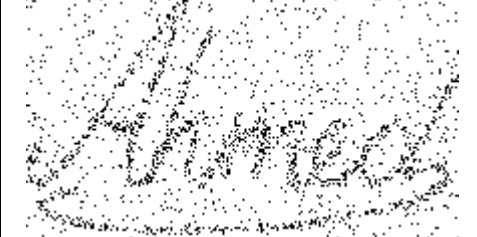


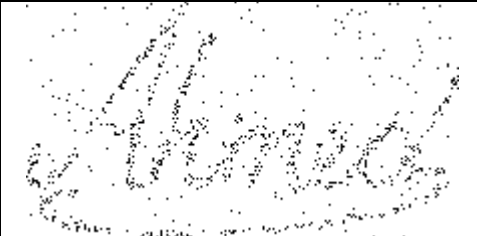
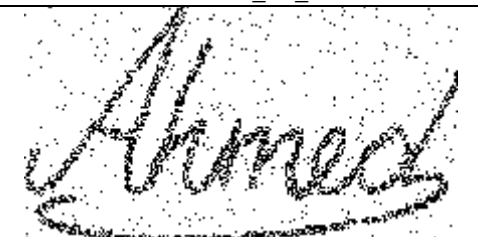
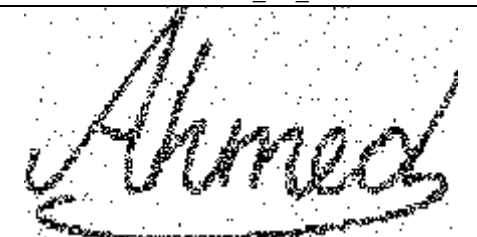
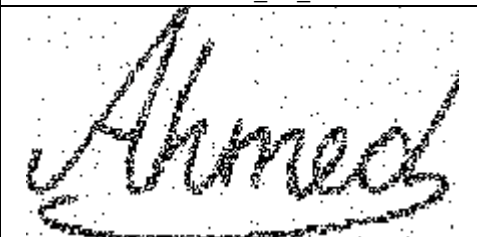
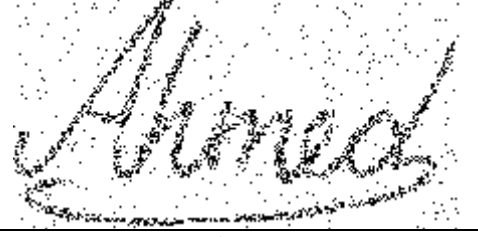
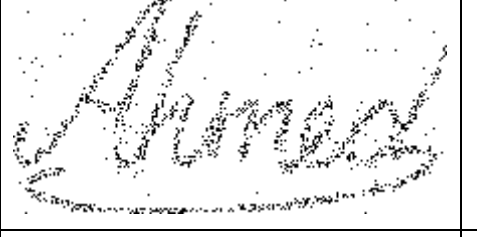


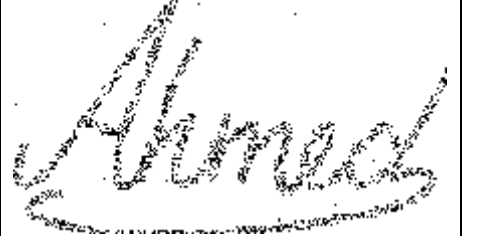
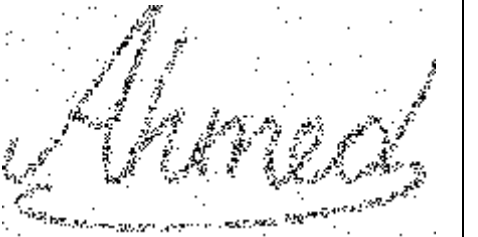
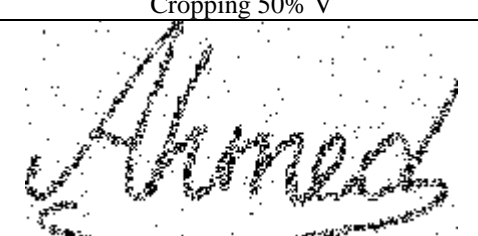
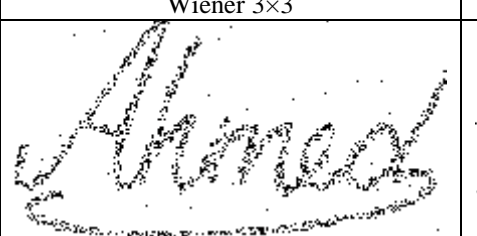
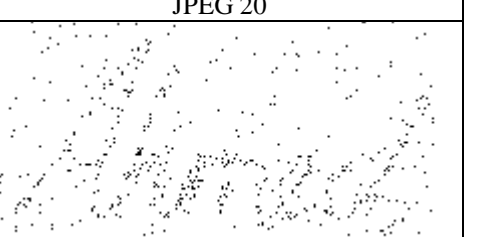
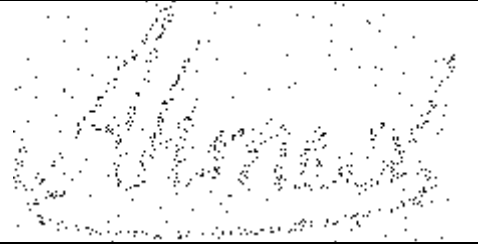


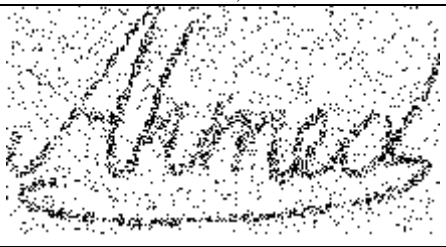
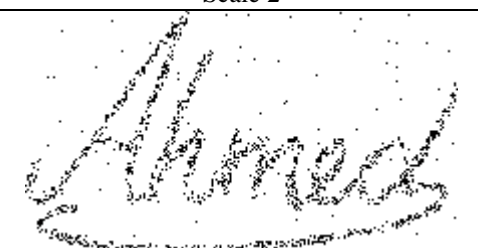
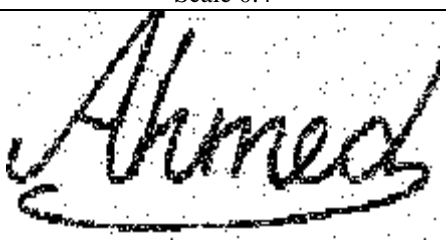
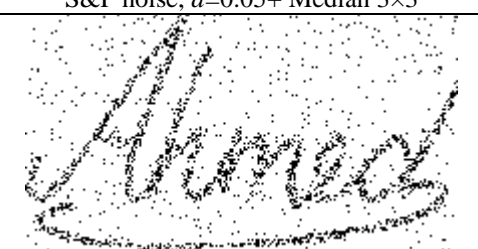
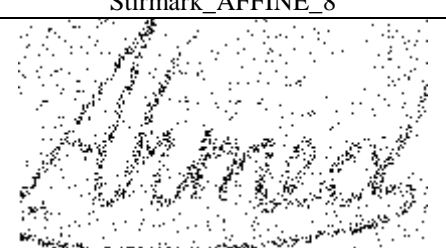
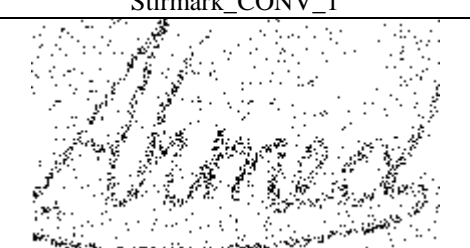
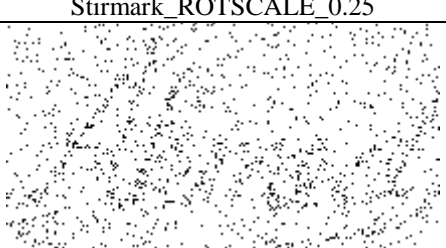
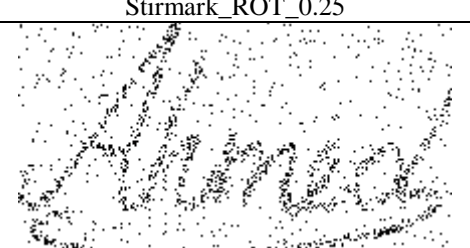
		
Stirmark_ROT_-0.5	Stirmark_ROT_CROP_-0.5	Stirmark_ROT_CROP_0.25
		
Stirmark_SS_1	Stirmark_SS_2	Stirmark_SS_3
		
Stirmark_RML_10	Stirmark_RML_50	Stirmark_RML_100

Table 4-25 Reconstructed watermarks after attacks at $\Delta = 34$

		
Cropping 75% V	Low pass 3x3	JPEG 75
		
Cropping 50% V	Wiener 3x3	JPEG 20
		
Cropping 75% H	Median 3x3	JPEG 15

		
Cropping 75% V+ H	S&P noise, $d=0.02$	Gaussian noise $m=0, v=0.002$
		
Scale 2	Scale 0.4	Gaussian noise $m=0, v=0.001$
		
S&P noise, $d=0.05$ + Median 3×3	Contrast enhancement intensity=0.3, 0.9	Contrast enhancement intensity=0.1, 0.5
		
StirMark_AFFINE_1	StirMark_AFFINE_8	StirMark_CONV_1
		
StirMark_ROTSCALE_-0.5	StirMark_ROTSCALE_0.25	StirMark_ROT_0.25
		
StirMark_ROT_-0.5	StirMark_ROTSCROP_-0.5	StirMark_ROTSCROP_0.25

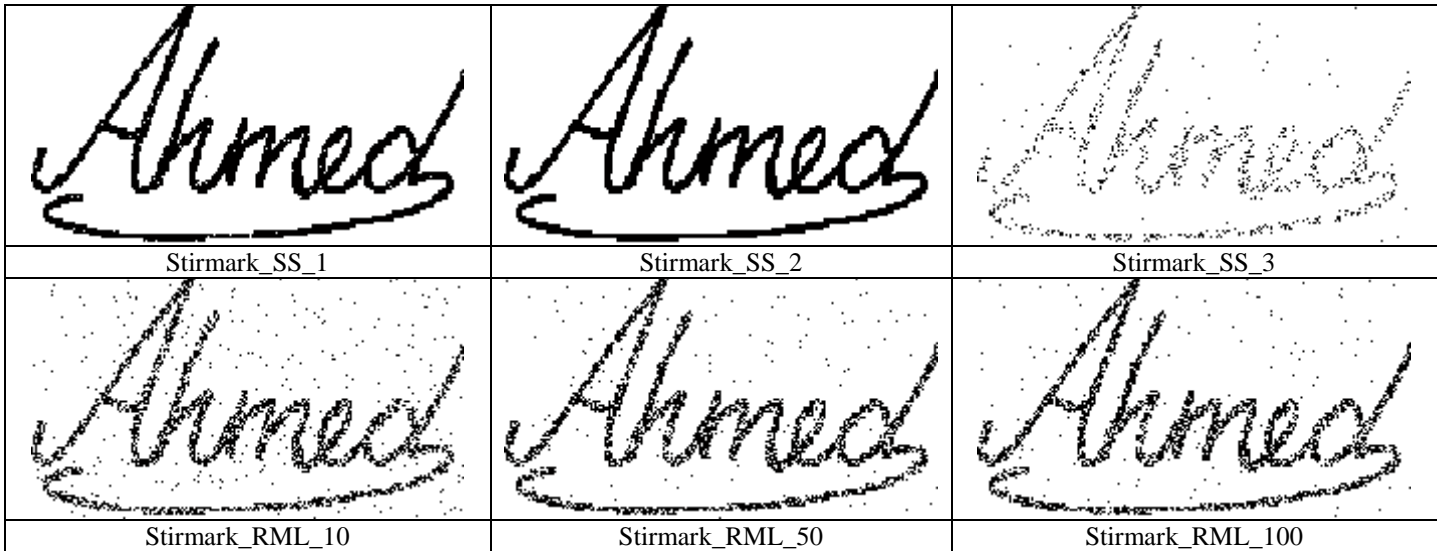


Table 4-26 Matlab execution time

Intel processor centrino 2 GHZ, RAM= 1 GB			
<i>Watermark size</i>	<i>Embedding time</i>	<i>Extraction time</i>	<i>Total time</i>
224 × 128	4.7969 seconds	6.8438 seconds	11.6407 seconds
224 × 96	4.7969 seconds	5.5781 seconds	10.3750 seconds
192 × 64	4.8438 seconds	3.1875 seconds	8.0313 seconds

Table 4-27 Performance evaluation against high resolution images

Watermark size 224×128 at $\Delta = 24$



(a): Original Host image



(b): Watermarked host image of size 1024×1024



(c): Watermarked host image of size 2048×2048

PSNR	SSIM	PSNR	SSIM
37.5926	0.9523	37.6645	0.9498
Attacks	host image size 1024×1024	host image size 2048×2048	
JPEG 30	NC= 1.0000	NC= 1.0000	
Cropping 75 % both sides	NC= 0.9914	NC= 0.9834	
Low-pass Filter 3×3	NC= 0.9980	NC=1.0000	
Median filter 3×3	NC= 0.9986	NC=1.0000	
Wiener filter 3×3	NC=1.0000	NC=1.0000	

4.5 Comparison with previous work

The Lena image is used in tables 4-28, 4-29 and 4-30 which represent comparisons between the 3 proposed colour algorithms and other watermarking methods [29, 31,

52, 58, 66]. The correlation coefficient (CC) is utilized to measure the similarity between the original watermark and the extracted ones. CC is used to compare the proposed method against the method in [29]. The algorithms introduced in [29] and [58] produce higher PSNR values than our proposed methods. However, higher PSNR values can be achieved in our methods by changing the watermarking strength value. It can be observed from Tables 4-28, 4-29 and 4-30 that the proposed algorithms generate higher NC and CC values and the extracted watermark is better quality compared to other watermarking methods, such as in [29, 31, 52, 58, 66]. Hence, it can be concluded that the proposed algorithms are more robust to attacks.

Table 4-28 Comparison between the proposed algorithms and other benchmark algorithms

Algorithm	Blind	Domain	Colour model	Host size	Watermark size	PSNR
[29]	No	Spatial	HIS-S	512×512	64×64	44.6
[31]	No	Spatial	RGB-B	512×512	32×32	38.9
[52]	Yes	DCT	RGB	512×512	64×64	37.6
[58]	Yes	DCT	YCbCr	512×512	64×64	41.1
[66]	Yes	DCT	YCbCr-Y	512×512	53×53	52
Algorithm 4	Yes	DCT	YCbCr-Y	512×512	64×64	39.8
Algorithm 5	Yes	DCT	RGB-G	512×512	64×64	40.9
Algorithm 6	Yes	DCT	RGB-G	512×512	224×128	38.5

Table 4-29 Comparison of robustness

Watermark size 64×64		Watermark size 64×64			
	Method in [29]	Algorithm 4		Algorithm 5	
Attacks	CC	CC Δ=12	CC Δ=16	CC Δ=24	CC Δ=34
JPEG 80	0.86	0.9745	0.9941	0.9751	0.9992
JPEG 70	0.78	0.9560	0.9898	0.9564	0.9916
JPEG 60	0.73	0.9325	0.9866	0.9350	0.9873
JPEG 50	0.68	0.9098	0.9830	0.9169	0.9843
Scale 2	0.97	1	1	1	1
Scale 0.5	0.92	0.7510	0.8625	0.7615	0.8770
Median filter 7×7	0.76	0.1599	0.1645	0.1622	0.1651
Cropping 25% side	0.65	0.9920	0.9920	0.9925	0.9925
Watermark size 64×64		Watermark size 64×64			
	Method in [31]	Algorithm 4		Algorithm 5	
Attacks	NC	NC Δ=12	NC Δ=16	NC Δ=24	NC Δ=34
JPEG 80	1	1	1	1	1
JPEG 75	0.82	1	1	1	1
JPEG 50	0.55	1	1	1	1
Median filter 3×3	1	0.9942	0.9985	1	1
Median Filter 5×5	1	0.9520	0.9710	0.9717	0.9912
Cropping 25% side	1	1	1	1	1
Cropping 75% H	1	1	1	1	1
Scale 2	1	1	1	1	1
Scale 0.5	1	0.9471	0.9899	0.9730	0.9950
Watermark size 64×64		Watermark size 64×64			
	Method in [52]	Algorithm 4		Algorithm 5	
Attacks	NC	NC Δ=12	NC Δ=16	NC Δ=24	NC Δ=34
JPEG 35	0.7906	0.9950	1	1	1
Gaus. noise m=0v=0.225	0.9810	0.5612	0.6571	0.6891	0.7014

S & P noise $d=0.03$	0.9545	0.9312	0.9550	0.9001	0.9234
Watermark size 64×64		Watermark size 64×64			
	Method in [58]	Algorithm 4		Algorithm 5	
Attacks	NC	NC $\Delta=12$	NC $\Delta=16$	NC $\Delta=24$	NC $\Delta=34$
Low-pass filter 3×3	0.9448	0.9862	0.9995	0.9923	1
Cropping 25%	0.9068	1	1	1	1
Cropping 50%	0.9563	0.9991	0.9991	0.9891	0.9891
S & P noise $d=0.0005$	0.9068	1	1	1	1
Watermark size 53×53		Watermark size 96×64			
	Method in [66]	Algorithm 4		Algorithm 5	
Attacks	NC	NC $\Delta=12$	NC $\Delta=16$	NC $\Delta=24$	NC $\Delta=34$
JPEG 80	0.97	1	1	1	1
JPEG 70	0.97	1	1	1	1
JPEG 60	0.91	1	1	1	1
JPEG 50	0.65	1	1	1	1
S & P noise $d=0.005$	0.86	0.9534	0.9677	0.9606	0.9932
S & P noise, $d=0.01$	0.81	0.8412	0.9345	0.8557	0.9606
S & P noise, $d=0.02$	0.72	0.6632	0.8541	0.6719	0.8606
Cropping 25%	0.72	0.9970	0.9972	0.9973	0.9973
Cropping 50%	0.50	0.9780	0.9780	0.9783	0.9783
Random cropping	0.80	0.9865	0.9866	0.9871	0.9871

Table 4-30 Comparison of robustness

Watermark size 224×128			
	Method in [29]	Algorithm 6	
Attacks	CC	CC $\Delta=24$	CC $\Delta=34$
JPEG 80	0.86	0.8209	0.9513
JPEG 70	0.78	0.7972	0.8734
JPEG 60	0.73	0.7434	0.8111
JPEG 50	0.68	0.7350	0.7953
Scale 2	0.97	1	1
Scale 0.5	0.92	0.4120	0.6412
Median filter 7×7	0.76	0.0710	0.1690
Cropping 25% side	0.65	0.6635	0.7863
	Method in [31]	Algorithm 6	
Attacks	NC	NC $\Delta=24$	NC $\Delta=34$
JPEG 80	1	0.9987	1
JPEG 75	0.82	0.9986	1
JPEG 50	0.55	0.9956	0.9989
Median filter 3×3	1	0.9968	0.9995
Median Filter 5×5	1	0.9872	0.9975
Cropping 25% side	1	0.9988	0.9988
Cropping 75% H	1	0.9909	0.9910
Scale 2	1	1	1
Scale 0.5	1	0.9428	0.9655
	Method in [52]	Algorithm 6	
Attacks	NC	NC $\Delta=24$	NC $\Delta=34$
JPEG 35	0.7906	0.9948	0.9980
Gaus. noise $m=0, v=0.225$	0.9810	0.5610	0.6811
S & P noise $d=0.03$	0.9545	0.8616	0.8711
	Method in [58]	Algorithm 6	
Attacks	NC	NC $\Delta=24$	NC $\Delta=34$
Low-pass filter 3×3	0.9448	0.9935	0.9986
Cropping 25%	0.9068	0.9964	0.9967
Cropping 50%	0.9563	0.9800	0.9801

S & P noise $d=0.0005$	0.9068	0.9815	0.9996
	Method in [66]	Algorithm 6	
Attacks	NC	NC$\Delta=24$	NC$\Delta=34$
JPEG 80	0.97	0.9987	1
JPEG 70	0.97	0.9982	0.9997
JPEG 60	0.91	0.9972	0.8111
JPEG 50	0.65	0.9956	0.7953
Salt & Pepper $d=0.005$	0.86	0.7836	0.8911
Salt & Pepper $d=0.01$	0.81	0.9444	0.9711
Salt & Pepper $d=0.02$	0.72	0.8889	0.9312
Cropping 25%	0.72	0.9988	0.9988
Cropping 50%	0.50	0.9910	0.9910
Random cropping	0.80	0.9919	0.9919

4.6 Final Remarks

In this chapter, different watermarking algorithms for colour images have been developed. Compared to other techniques, the experimental results show that the proposed colour algorithms have excellent invisibility qualities and were found to be robust against JPEG compression up to 15% quality, cropping, small degrees of rotation up to 0.5 degree, scaling, additive noise, filtering operations and Stirmark attacks. The first watermarking algorithm embedded the watermark into the Y channel of the host colour image by selectively modifying the very low frequency components of the DCT. The second algorithm used the green channel for embedding the watermark. The green channel has been chosen after carrying out analytical experiments using some popular measurement metrics. The green channel for watermark embedding has proven to provide excellent invisibility qualities with strong robustness. The third algorithm is a high capacity technique that embeds watermarking information using 25% of the host image size. The maximum numbers of bits that can be hidden and recovered successfully from the third watermarking algorithm has been increased.

The proposed watermarking algorithms in this chapter can be used in different applications. In most applications the watermarking algorithm must embed the watermark in such a way that it does not affect the quality of the host data. As demonstrated in section 4.5, the proposed methods provide excellent invisibility qualities compared to other watermarking methods. For copyright protection the owner can embed the watermark representing copyright information without affecting the quality of original data. This can prove the ownership in court when someone infringed on the copyrights. For the protection of intellectual property

rights, it seems reasonable that one wants to embed an amount of information similar to the year of copyright, an ISBN (International Standard Book Numbering) number or dedicated barcode.

The proposed algorithms can be used for transmission of secret private messages. Medical safety and patient privacy is important in medical fields. The proposed methods can be used for embedding the date and patient's name in medical images. Handwritten signatures were used as watermarks through out the implemented algorithms in this chapter. Mobile phone numbers will be used as watermarks in the next chapter.

Chapter 5 Watermarking Algorithms for Images Captured by Mobile Phone Cameras

5.1 Overview

This chapter investigates digital watermarking algorithm as a solution to the problem of copyright protection of digital images captured by mobile phone camera. The aim is to develop a technique that will make it possible to verify the authenticity of mobile camera photos. The developed watermark algorithm here uses mobile phone numbers with the international country code as watermark. This algorithms is carried out using the low frequency coefficients of the DCT. The presented algorithm embeds the watermarks in the Green component of the RGB of the colour image.

5.2 Algorithm 7: Blind Image Watermarking of Mobile Phone Numbers Using Low-Frequency DCT Coefficients

A new frequency domain based watermarking scheme for colour images captured by mobile phone cameras is proposed. The proposed technique embeds personal mobile phone numbers inside the image. The aim of the scheme is to protect the copyright ownership of the image. Each bit of the decimal digits is inserted onto one low frequency coefficient of one of the DCT blocks of the host image. A DCS process was applied to increase the invisibility qualities. This process managed to search the DCT sub-block excluding the DC coefficient to find the coefficients with the maximum magnitude. Different embedding locations, depending on the spatial frequencies of the host image, will be selected. The proposed algorithm achieves a high PSNR values and is found to be robust against JPEG compression and different image manipulation algorithms.

One of the key problems facing the use of mobile phone numbers to authenticate digital images or to investigate the integrity is that embedding the number in an image has special requirements. Visually recognizable patterns are more intuitive and easier for representing one's identity even if the watermarked image undergo any attacks that will affect the quality of the extracted watermark. On the other hand, this characteristic is not available when using decimal numbers for representing one's identity. Mobile phone numbers are required to be intact and survive strong attacks. Otherwise, it will be very difficult to recognise the mobile phone number. This means that for the proposed technique to be successful it should not only provide good performance in terms of PSNR, SSIM and NC, but also be able to survive strong and higher levels of attacks such as, JPEG, and filtering using a 5×5 mask size.

5.2.1 The Proposed Algorithm

For this work, the phone number plus the international country code is used as the watermark. The summation of the decimal digits is added to the number to make it 16 decimal digits. This is useful to check if the extracted number is correct or not. A special procedure is applied if the summation exceeds 99. For example, the maximum summation that can be achieved is 126 when a mobile phone number with 14 digits, where all the digits have a value of nine, is entered. In this case a split process will be implemented and the check sum digits as shown in Figure 5.1 will hold 12 and 6 instead of 6. This is made possible because 4-bit Hexadecimal code is used to represent each digit, which generates 64 binary bits. Figure 5.1 shows an example of a UAE mobile number.

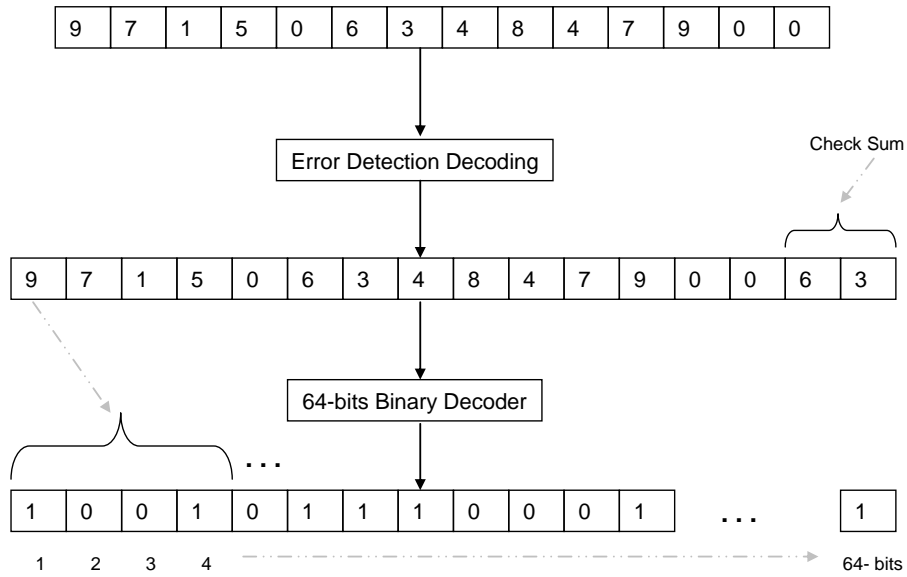


Figure 5-1 Graphical presentation for Error and binary Decoders

5.2.2 The DCT Selection Coefficients (DCS) Process

The DCT block consists of 8×8 coefficients. The 16 lower frequencies are screened to find the coefficient with the highest magnitude and register its location. This process is repeated for all the DCT blocks. The location which is repeated most is selected and saved to be used in the extraction process. This location will vary from one image to another according to the spatial frequency contents of the image. One binary bit of the watermark will be embedded at this location. A flow graph of the DCS process, which is used to select one coefficient, is shown in Figure 5.2. Table 5.1 represents the registered location of some images. In order to test the security of the DCS process, the images were screened again after embedding to verify that the method is secure and an attacker would not be able to use the DCS process to detect the originally selected locations. Screening the DCT blocks again after embedding will result in totally different locations from the previously registered locations in the original images as shown in Table 5.2. All the natural scene images in Table 5.1 were recaptured using a mobile phone camera.

Table 5-1 DCS Locations for Original un-watermarked images

























The DCS locations for some images					
<i>Image</i>	<i>Coefficient</i>	<i>Image</i>	<i>Coefficient</i>	<i>Image</i>	<i>Coefficient</i>
	(1,2)		(1,2)		(2,1)
	(1,2)		(2,1)		(2,1)
	(1,2)		(2,1)		(2,1)
	(2,1)		(2,1)		(1,2)

Table 5-2 DCS Locations for watermarked images

The DCS locations for some images					
<i>Image</i>	<i>Coefficient</i>	<i>Image</i>	<i>Coefficient</i>	<i>Image</i>	<i>Coefficient</i>
	(2,1)		(2,1)		(1,2)
	(2,1)		(1,2)		(1,2)
	(2,1)		(1,2)		(3,1)
	(3,1)		(3,1)		(2,1)

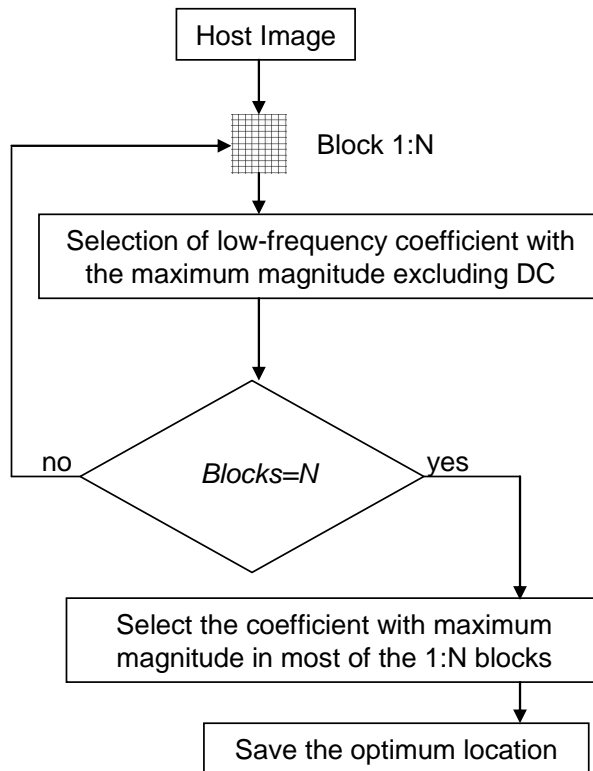


Figure 5-2 A flow graph of the DCS process

5.2.3 Embedding and Extraction Steps

The proposed watermarking scheme is based on the possibility of embedding multiple copies of the same mobile number in the host image. The embedding algorithm here is totally blind. The watermark data is embedded in the very low-DCT frequency component obtained from the DCS process. Inside each 8×8 sub-blocks, one DCT coefficient is identified and used for the embedding process. The predefined coefficient has been obtained from the DCS process applied previously to the host image. It is worth mentioning that for each host image a different predefined coefficient will be selected. Thus, the invisibility qualities will be increased. The binary mobile number digits are randomly scrambled using a secret key. This scrambling process is essential to reduce the spatial correlation between the host image and the embedded watermark. After the scrambling process, the shuffle scheme is applied for each copy of the binary mobile number to shuffle the binary digits before the embedding process. All the watermarking embedding steps are described in Figure 5.3. It is important to note that the watermark is embedded several times in the host image depending on the sizes of the host and watermark images.

The embedded watermark information can be extracted by performing an 8×8 DCT transform for the watermarked host image and then indicating the same coefficient of the host image that carries the bits of the embedded watermarks using the required secret key. It is worth mentioning that although the proposed scheme is blind, it requires information such as the sizes of both the host and watermark images and the watermark embedding strength Δ . The extraction formula defined in chapter 3 is used to produce the scrambled watermark. According to the key in the initial scrambling operation, the scrambled watermark is descrambled to retrieve the original watermark. A reverse shuffling scheme is implemented for each reconstructed watermark. Simply, the recovery function is the inverse of all the watermarking embedding steps. The watermarking extraction steps are shown in Figure 5.4.

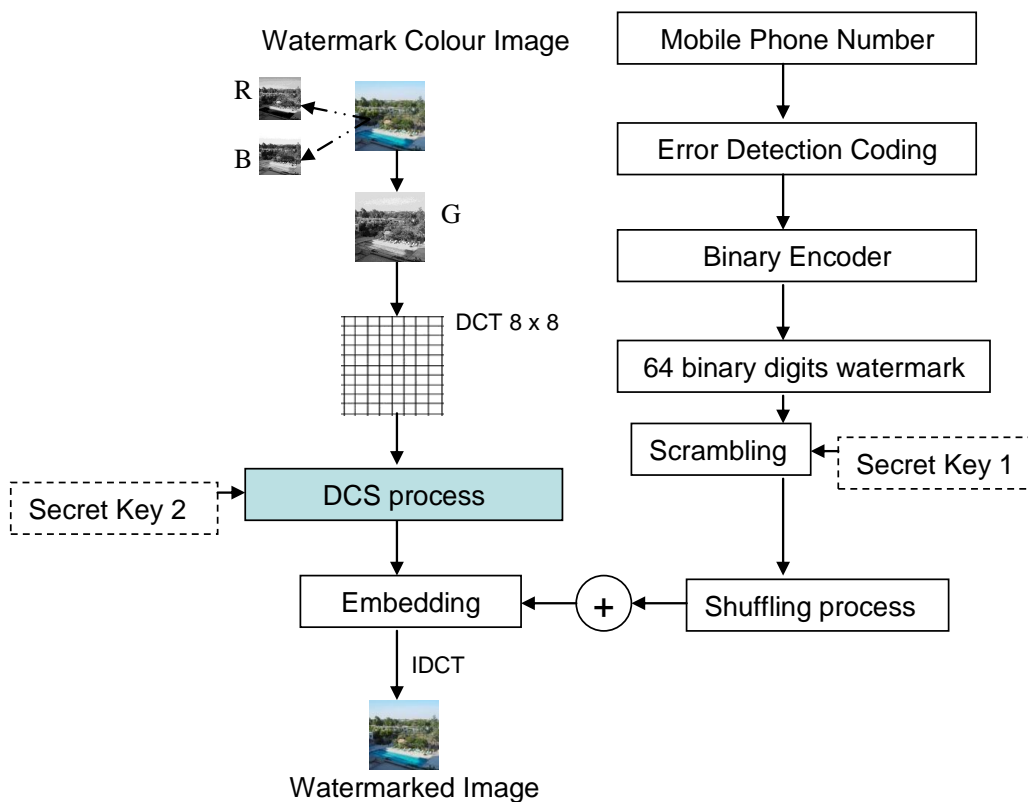


Figure 5-3 Graphical presentation for embedding steps

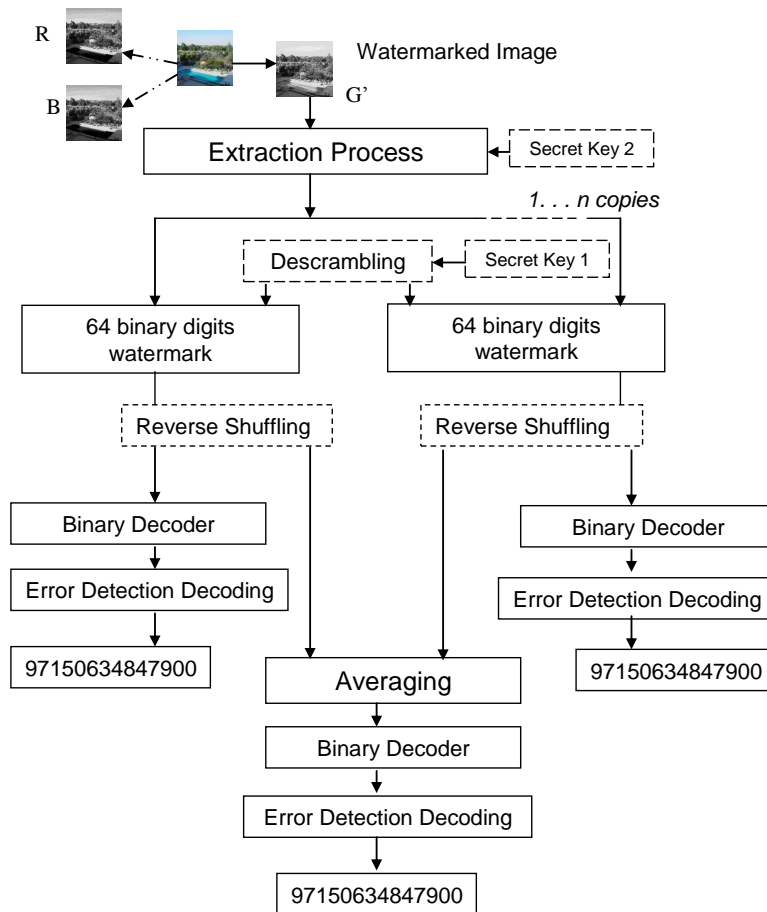


Figure 5-4 Graphical presentation for extraction steps

5.2.4 Results

The perceptual invisibility is evaluated using PSNR at different embedding strengths as shown in Table 5.3. The PSNR values between the original “Lena” and the watermarked images are 90.4 dB and 78.1 dB for watermarking strengths $\Delta = 16$ and $\Delta = 40$, respectively. In Table 5.4 the perceptual invisibility of the proposed algorithm is evaluated using SSIM at different embedding strengths. The original “Lena” image has been used to examine the perceptual quality at different embedding strengths as depicted in Table 5.5.

To verify the robustness of the proposed method, various common signal processing and geometric attacks are applied to the watermarked images. NC is used to measure the similarity between the original and the extracted watermark. Table 5.6 demonstrates the performance of the proposed method when using the country code and a mobile phone number from the United Arab Emirates (UAE), Egypt and Finally, the United Kingdom (UK) at watermarking strength $\Delta = 24$.

The experimental results show that the performance achieved by the proposed method for the extracted watermark after running different attacks is perceptually visible when $\Delta = 24$, which can be considered as the best value for the embedding strength. Higher embedding strength value such as $\Delta = 3$ and 4 as shown in Table 5.7, will provide strong robustness and distinct perceptual visibility for the extracted watermark. It is worth noting that higher embedding strength could reduce the invisibility qualities as demonstrated in Table 5.3. Special requirements are needed when using mobile phone numbers as a method to authenticate digital images. From Tables 5.3 and 5.4 normalized correlation values that are less than one indicates that the algorithm has failed to restore the embedded data.

The Matlab execution time of the proposed algorithm is shown in Table 5.8. Finally, the performance evaluation against high resolution images is illustrated in Table 5.9.

Table 5-3 PSNR for different colour images

Peak Signal to Noise Ratio			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
PSNR at $\Delta = 16$	51.4395	50.6715	54.5712
PSNR at $\Delta = 24$	47.8758	46.5765	49.1367
PSNR at $\Delta = 34$	44.7981	43.1761	46.8171
PSNR at $\Delta = 40$	43.2219	42.0291	45.7610

Table 5-4 SSIM for different colour images

Structural Similarity Index Measurements			
<i>Image</i>	<i>Lena</i>	<i>Pepper</i>	<i>Baboon</i>
SSIM at $\Delta = 16$	0.9979	0.9963	0.9993
SSIM at $\Delta = 24$	0.9950	0.9918	0.9985
SSIM at $\Delta = 34$	0.9902	0.9829	0.9970
SSIM at $\Delta = 40$	0.9865	0.9778	0.9957

Table 5-5 Original and watermarked Lena images at different embedding strengths

	
Original un-watermarked Lena image	
	
Watermarked image at $\Delta = 16$	Watermarked image at $\Delta = 24$
	
Watermarked image at $\Delta = 34$	Watermarked image at $\Delta = 40$

Table 5-6 Normalized correlation for Lena colour image at $\Delta = 24$

NC values at $\Delta = 24$

United Arab Emirates (UAE) mobile number = 97150634847900			
Attacks	NC	Attacks	NC
Cropping 50% V	0	Low pass 3×3	1
Cropping 48% V	1	Low pass 5×5	1
Cropping 75% H	1	Wiener 3×3	1
Cropping 50% H	1	Wiener 5×5	1
Gaussian noise m=0, v=0.002	0	Median 3×3	1
Gaussian noise m=0, v=0.001	1	Median 5×5	1
S&P noise, d=0.02+ Median 3×3	1	JPEG 50	1
S&P noise, d=0.05+ Median 3×3	1	JPEG 25	1
Contrast enhancements intensity=0.3, 0.9	1	JPEG 18	1
Scale 2	1	Scale 0.4	1
Stirmark_AFFINE_1	1	Stirmark_CONV_1	0
Stirmark_AFFINE_8	0	Stirmark_RML_10	1
Stirmark_ROTSCALE_0.25	1	Stirmark_RML_100	1
Stirmark_ROTSCALE_-0.5	0	Stirmark_SS_1	1
Stirmark_ROT_0.25	1	Stirmark_SS_2	1
Stirmark_ROT_-0.5	0	Stirmark_SS_3	1
Stirmark_ROTSCROP_-0.5	0	Stirmark_ROTSCROP_0.25	1
NC values at $\Delta = 24$ Egypt mobile number =2012333603800			
Attacks	NC	Attacks	NC
Cropping 50% V	0	Low pass 3×3	1
Cropping 48% V	1	Low pass 5×5	1
Cropping 75% H	1	Wiener 3×3	1
Cropping 50% H	1	Wiener 5×5	1
Gaussian noise m=0, v=0.002	0	Median 3×3	1
Gaussian noise m=0, v=0.001	1	Median 5×5	1
S&P noise, d=0.02+ Median 3×3	1	JPEG 50	1
S&P noise, d=0.05+ Median 3×3	1	JPEG 25	1
Contrast enhancements intensity=0.3, 0.9	1	JPEG 18	1
Scale 2	1	Scale 0.4	1
Stirmark_AFFINE_1	1	Stirmark_CONV_1	0
Stirmark_AFFINE_8	0	Stirmark_RML_10	1
Stirmark_ROTSCALE_0.25	1	Stirmark_RML_100	1
Stirmark_ROTSCALE_-0.5	0	Stirmark_SS_1	1
Stirmark_ROT_0.25	1	Stirmark_SS_2	1
Stirmark_ROT_-0.5	0	Stirmark_SS_3	1
Stirmark_ROTSCROP_-0.5	0	Stirmark_ROTSCROP_0.25	1
NC values at $\Delta = 24$ United kingdom (UK) mobile number=44772070772400			
Attacks	NC	Attacks	NC
Cropping 75% V	0	Low pass 3×3	1
Cropping 48% V	1	Low pass 5×5	1
Cropping 75% H	1	Wiener 3×3	1
Cropping 50% H	1	Wiener 5×5	1
Gaussian noise m=0, v=0.002	0	Median 3×3	1
Gaussian noise m=0, v=0.001	1	Median 5×5	1
S&P noise, d=0.02+ Median 3×3	1	JPEG 50	1
S&P noise, d=0.05+ Median 3×3	1	JPEG 25	1
Contrast enhancements intensity=0.3, 0.9	1	JPEG 18	1
Scale 2	1	Scale 0.4	1
Stirmark_AFFINE_1	1	Stirmark_CONV_1	0
Stirmark_AFFINE_8	0	Stirmark_RML_10	1
Stirmark_ROTSCALE_0.25	1	Stirmark_RML_100	1
Stirmark_ROTSCALE_-0.5	0	Stirmark_SS_1	1
Stirmark_ROT_0.25	1	Stirmark_SS_2	1
Stirmark_ROT_-0.5	0	Stirmark_SS_3	1

StirMark_ROT_CROP_-0.5	0	StirMark_ROT_CROP_0.25	1
------------------------	---	------------------------	---

Table 5-7 Normalized correlation for Lena colour image at $\Delta = 34$

NC values at $\Delta = 34$			
United Arab Emirates (UAE) mobile number = 97150634847900			
Attacks	NC	Attacks	NC
Cropping 50% V	0	Low pass 3x3	1
Cropping 48% V	1	Low pass 5x5	1
Cropping 75% H	1	Wiener 3x3	1
Cropping 50% H	1	Wiener 5x5	1
Gaussian noise m=0, v=0.002	1	Median 3x3	1
Gaussian noise m=0, v=0.001	1	Median 5x5	1
S&P noise, d=0.02+ Median 3x3	1	JPEG 50	1
S&P noise, d=0.05+ Median 3x3	1	JPEG 25	1
Contrast enhancements intensity=0.3, 0.9	1	JPEG 18	1
Scale 2	1	Scale 0.4	1
StirMark_AFFINE_1	1	StirMark_CONV_1	0
StirMark_AFFINE_8	0	StirMark_RML_10	1
StirMark_ROT_SCALE_0.25	1	StirMark_RML_100	1
StirMark_ROT_SCALE_-0.5	0	StirMark_SS_1	1
StirMark_ROT_0.25	1	StirMark_SS_2	1
StirMark_ROT_-0.5	0	StirMark_SS_3	1
StirMark_ROT_CROP_-0.5	0	StirMark_ROT_CROP_0.25	1
NC values at $\Delta = 34$			
Egypt mobile number =2012333603800			
Attacks	NC	Attacks	NC
Cropping 50% V	0	Low pass 3x3	1
Cropping 48% V	1	Low pass 5x5	1
Cropping 75% H	1	Wiener 3x3	1
Cropping 50% H	1	Wiener 5x5	1
Gaussian noise m=0, v=0.002	1	Median 3x3	1
Gaussian noise m=0, v=0.001	1	Median 5x5	1
S&P noise, d=0.02+ Median 3x3	1	JPEG 50	1
S&P noise, d=0.05+ Median 3x3	1	JPEG 25	1
Contrast enhancements intensity=0.3, 0.9	1	JPEG 18	1
Scale 2	1	Scale 0.4	1
StirMark_AFFINE_1	1	StirMark_CONV_1	0
StirMark_AFFINE_8	0	StirMark_RML_10	1
StirMark_ROT_SCALE_0.25	1	StirMark_RML_100	1
StirMark_ROT_SCALE_-0.5	0	StirMark_SS_1	1
StirMark_ROT_0.25	1	StirMark_SS_2	1
StirMark_ROT_-0.5	0	StirMark_SS_3	1
StirMark_ROT_CROP_-0.5	0	StirMark_ROT_CROP_0.25	1
NC values at $\Delta = 34$			
United kingdom (UK) mobile number=44772070772400			
Attacks	NC	Attacks	NC
Cropping 75% V	0	Low pass 3x3	1
Cropping 48% V	1	Low pass 5x5	1
Cropping 75% H	1	Wiener 3x3	1
Cropping 50% H	1	Wiener 5x5	1
Gaussian noise m=0, v=0.002	1	Median 3x3	1
Gaussian noise m=0, v=0.001	1	Median 5x5	1
S&P noise, d=0.02+ Median 3x3	1	JPEG 50	1
S&P noise, d=0.05+ Median 3x3	1	JPEG 25	1
Contrast enhancements intensity=0.3, 0.9	1	JPEG 18	1
Scale 2	1	Scale 0.4	1

Stirmark_AFFINE_1	1	Stirmark_CONV_1	0
Stirmark_AFFINE_8	0	Stirmark_RML_10	1
Stirmark_ROTSCALE_0.25	1	Stirmark_RML_100	1
Stirmark_ROTSCALE_-0.5	0	Stirmark_SS_1	1
Stirmark_ROT_0.25	1	Stirmark_SS_2	1
Stirmark_ROT_-0.5	0	Stirmark_SS_3	1
Stirmark_ROTSCROP_-0.5	0	Stirmark_ROTSCROP_0.25	1

Table 5-8 Matlab execution time

Intel processor centrino 2 GHZ, RAM= 1 GB			
<i>Watermark size</i>	<i>Embedding time</i>	<i>Extraction time</i>	<i>Total time</i>
16 digits mobile no.	2.4063 seconds	1.9688 seconds	4.3751 seconds

Table 5-9 Performance evaluation against high resolution images



(a): Original Host image



(b): Watermarked host image of size 1024×1024



(c): Watermarked host image of size 2048×2048

PSNR	SSIM	PSNR	SSIM
47.3917	0.9929	47.2748	0.9929
Attacks	host image size 1024×1024	host image size 2048×2048	
JPEG 30	NC=1.0000	NC=1.0000	
Cropping 75 % both sides	NC=1.0000	NC=1.0000	
Low-pass Filter 3×3	NC=1.0000	NC=1.0000	
Median filter 3×3	NC=1.0000	NC=1.0000	
Wiener filter 3×3	NC=1.0000	NC=1.0000	

5.3 Comparison with Previous Work

Table 5.8 represents comparisons between the proposed algorithm and the watermarking method reported in [72]. The number of error bits is used as the basis

for this comparison. It can be observed from Table 5.8 that the extracted watermarks are bits error free compared to the watermarking methods in [72]. Hence, the proposed algorithm is more robust to attacks.

Table 5-10 Normalized correlation for Lena colour image at $\Delta=34$

United kingdom (UK) mobile number=44772070772400				
Attacks	Method in [72]		Algorithm 7	
	Error bits Test image 1	Error bits Test image 2	Error bits at $\Delta=24$	Error bits at $\Delta=34$
JPEG 100	0	0	0	0
JPEG 90	0	0	0	0
JPEG 80	0	0	0	0
JPEG 70	0	0	0	0
JPEG 60	1	0	0	0
JPEG 50	1	0	0	0
JPEG 40	11	8	0	0
Median filter 3×3	3	12	0	0
Low-pass 3×3	2	10	0	0

5.4 Final Remarks

In this chapter, a novel watermarking scheme for colour images captured by mobile phone cameras is proposed. The proposed technique embeds personal mobile phone numbers including the international code of the country inside the image. The aim of the scheme is to protect the copyright ownership of the image. Each bit of the decimal digits is inserted onto one low frequency coefficient of one of the DCT blocks of the host image. A DCS process has been applied to increase the invisibility qualities. This process finds the coefficient with maximum magnitude. Different embedding locations are selected depending on the spatial frequencies of the host image. The proposed algorithm achieves a very high PSNR values and was found to be robust against JPEG compression, cropping up to 50%, small degrees of rotation up to 0.25%, scaling down up to 60%, additive noise, filtering operations and Stirmark attacks.

Chapter 6 Conclusions and Recommendations

6.1 Overview

Digital image watermarking, like any new area of research, has many drawbacks and challenges. Many researchers have proposed their solutions to solve some of the problems related to watermarking. The technology of watermarking needs many enhancements to be legally acceptable in courts and for proof of rightful ownership. Hundreds of points of views and approaches have been proposed and described in the literature, with several stories of success. However, this technology needs several years and a lot of extensive work in several areas related to it being standardized for use. This chapter presents a summary of the work carried out, highlighting the main conclusions and giving some recommendations for future work.

6.2 Summary of the work

The main achievements of the research presented in this thesis can be described as the development and evaluation of blind Discrete Cosine Transform-Based watermarking algorithms for copyright protection of digital images using handwritten signatures and mobile phone numbers. The watermark is embedded in the low-frequencies DCT coefficients. This range of frequencies was chosen because the high frequency components may be discarded in some image processing operation such as JPEG compression. A shuffle scheme was applied for each binary watermark copy before embedding by representing the watermark in a vector format and applying a shift operation to this vector. The shuffle scheme is necessary to reduce the spatial correlation between the watermark and the image. This has managed to increase the robustness against vertical cropping attacks. The watermark is further protected by using a secret key. The algorithm used is blind and does not require the original image for extracting the watermark. Multiple copies of the same signature are embedded in the host image. This increases the robustness of the watermark against several attacks since each watermark is individually reconstructed and verified before applying an averaging process. The quality of the reconstructed

signature was evaluated by using normalized correlation and correlation coefficient factors. The new watermarking algorithms cause little distortion to the host images so as to be invisible. The watermarking methods were shown to be robust against JPEG compression, additive noise, cropping, scaling, low-pass and median filtering, and Stirmark attacks. The algorithms have been examined using 35 different colour images of size 512×512 and also high resolution images of sizes 1024×1024 and 2048×2048 . Two evaluation techniques were used in the experiment with different watermarking strengths and different signature sizes. The PSNR and the structural similarity index measurement (SSIM) between the host image and the watermarked image were used. The higher the SSIM percentage is, the larger the similarity between the compared images. The performances of the algorithms were compared to other schemes reported in the literature to highlight the advantages of our proposed technique.

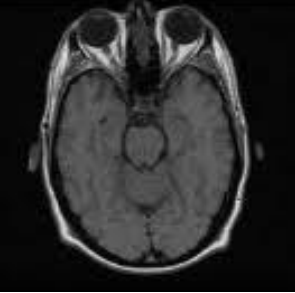
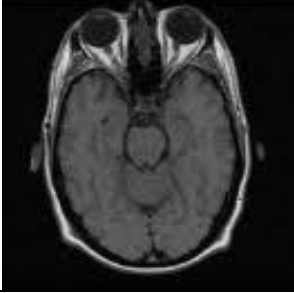
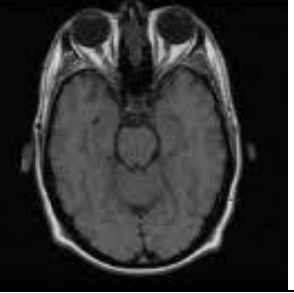





6.3 Conclusions

The concluding remarks of the developed techniques can be described in the following:

- The designed watermarking systems in this research deal easily with different file formats, different sizes and type of images as well as different sizes and types of watermarks. The algorithms have been examined using different images of size 512×512 bits per pixel and also high resolution images of size 1024×1024 and 2048×2048 . In this research, handwritten signatures and mobile phone numbers have been used as watermarks rather than the conventional pseudo random numbers. Handwritten signatures are more intuitive and easier for representing one's identity. Even if the watermarked image undergoes any attack that will affect the quality of the extracted watermark, the extracted watermarks can be recognizable with some errors. On the other hand, working with decimal numbers (phone numbers) as watermarks found to be more difficult, since a single bit error will lead to a totally different and incorrect number. In order to verify the findings, different types of host images such as medical images and different kind of watermarks such as barcodes, faces and text have been tested and verified; samples are shown in Table 6-1. At the same time, the dimensions of the host images and the watermarks in the proposed

methods are limited to powers of 2. However, if the host image dimensions are not powers of 2 they can be resized by padding with zero pixels to make it so.

Table 6-1 Samples of other tested host and watermark images

		
Original Un-watermarked image	Watermarked image at $\Delta = 12$	Watermarked image at $\Delta = 16$
		
Original Un-watermarked image	Watermarked image at $\Delta = 12$	Watermarked image at $\Delta = 16$
		Ahmed ALgindy 14-5-2007 Ajman University UAE
Barcode watermark	Face watermark	Text watermark

- It has been found that the designed methods are more robust and outperform most of the techniques reported in the literature. This is due to the use of multiple embedding which allows multiple watermarks to be inserted in an image, with each watermark still being independently verifiable. The use of the averaging process made it possible to reduce the amount of errors in the extracted watermarks. Multiple embedding-extraction for a single watermark is uncommon in most of the watermarking algorithms in the literature. Another hidden advantage of the multiple embedding process is that an attacker cannot compare several watermarked images and discover the modified patterns. This is because the watermarks are distributed all over the host image. At the same time, the multiple embedding-extraction process will increase the time required for embedding and extraction especially in high resolution images.

- It has been proven that the proposed image watermarking algorithms are effective in satisfying the major watermarking requirements and moreover, they can be easily implemented in real time. The choice of the discrete cosine transform has been found to be theoretically successful to reduce the computation complexity, since all the mobile phone cameras and digital cameras utilize the 8×8 DCT blocks for JPEG compression, This will simplify the future real time implementation of the proposed algorithms. Texas Instruments OMAP (Open Multimedia Application Platform) is a category of microprocessors that has capabilities for portable and mobile multimedia applications and is developed by Texas Instruments. Nokia, Samsung and Sony mobile phones use OMAP3 processors for multimedia and image processing purpose.
- A shuffle scheme, applied to each watermark before embedding, was included in all the watermarking algorithms presented in this thesis and was also compared to the case of no shuffle scheme. It was found that techniques with shuffle scheme were more robust against cropping attacks. The shuffle scheme manages to maintain the blindness of the implemented techniques. At the same time, the method can be applied only when multiple-embedding process is used. For example, the proposed method in [24] cannot survive any vertical cropping attacks because of the spatial correlation between the host image sub-blocks and the sub-blocks of the watermark copies. It has been proven that applying the shuffle method to the technique in [24] has excellent influence in increasing the robustness against cropping vertical attacks.
- It has been found that the use of the DCS process has increased the security of the watermarking algorithms. The attacker would not be able to use the DCS process again to detect the originally selected locations. A DCT coefficient(s) selection process (DCS) has been developed to increase the security of algorithms (2, 4, 5 and 7) and to reduce the visual changes when viewed by human eyes. This process has managed to observe the perceptual capacity of the low frequency coefficients inside each of the DCT blocks to select best coefficient(s). At the same time, the process will increase the time required for embedding. The process is designed only to select the coefficients with the maximum magnitude among the low frequency ones. The DCS can be modified to select the coefficients with the maximum magnitude among each 8×8 sub-block, but this will result in slowing down the embedding process.

- Colour watermarking techniques (algorithms 4 and 5) have been proposed. Both techniques were evaluated using PSNR and SSIM methods. The perceptual invisibility of the proposed algorithms was demonstrated at different embedding strengths. It can be concluded that using the green channel (algorithm 5) provides better perceptual invisibility and stronger robustness than using the Y component (algorithm 4) of the YCrCb model. Moreover, it has been proven by experimental measurements that the green channel is more suitable for watermarking embedding compared to the red and blue channels in the RGB model. It has been concluded also that, in algorithm 4 the watermarking strengths is much smaller than the ones used in algorithm 5. It has been found also that the green channel has the highest watermarking strengths delta, perceptual invisibility and stronger robustness among all other algorithms in this research.

- High capacity watermarking techniques (algorithms 3 and 6) have been proposed. Both algorithms were examined using different images of size 512×512 . Hand written signatures of size 224×128 and 192×64 were used as watermarks. The use of algorithms 3 and 6 resulted in increasing the ability of hiding information up to 25% of the host image size while maintaining the robustness and the perceptual invisibility. At the same time, the total processing time for the techniques (3 and 6) are largest among the techniques in this research. This is due to large watermark sizes. It must be noticed that the DCS process has not been applied to the high capacity algorithms (3 and 6). After applying a zigzag process, the first 16 low frequencies coefficients have been used excluding the DC coefficient. Table 6.2 presents the total processing time and the perceptual invisibility PSNR values using the same watermark size and watermarking strength for all the algorithms except algorithm 7.

Table 6-2 Comparison of the proposed algorithms

Comparison of the proposed algorithms					
<i>Algorithms</i>	<i>Watermark size</i>	<i>Number of DCT coefficients</i>	Δ	<i>Processing time</i>	<i>PSNR</i>
Algorithm1	96×64	8	14	4.01 seconds	40.2738 dB
Algorithm2	96×64	8	14	4.01 seconds	40.2738 dB
Algorithm3	96×64	16	14	6.41 seconds	37.7366 dB
Algorithm4	96×64	8	14	4.58 seconds	38.7101 dB
Algorithm5	96×64	8	14	4.59 seconds	44.9037 dB
Algorithm6	96×64	16	14	7.02 seconds	42.7610 dB
Algorithm7	16 digits mobile no.	1	14	4.37 seconds	52.6359 dB

- It has been found that different watermarking algorithms and different watermarking colour channels as well as different applications require different values of watermark strength Δ . An increase in the watermark embedding strength increases the distortion in the watermarked images. The recommended figures for the watermark strengths Δ for each developed algorithm concluded from this research will help developers in the future in implementing some of their algorithms in real time. Table 6.2 shows the PSNR values at the maximum recommended watermarking strengths Δ for the different watermarking algorithms used in this research.

Table 6-3 Perceptibility vs. maximum watermarking strengths

Perceptibility vs. watermarking strengths				
<i>Algorithms</i>	<i>Watermark size</i>	<i>Recommended</i>	<i>Max. recommended</i>	<i>PSNR at max</i>
		Δ	Δ	Δ
Algorithm1	96×64	14	16	39.2852 dB
Algorithm2	96×64	14	16	39.2852 dB
Algorithm3	224×128	14	16	36.9027 dB
Algorithm4	96×64	12	16	37.8383 dB
Algorithm5	96×64	24	34	38.2083 dB
Algorithm6	224×128	24	34	35.8011 dB
Algorithm7	16 digits mobile no.	24	34	44.7981 dB

- It has been found that the use of mobile phone numbers plus the international code is an easy and informative way of protecting the copyright of images captured by mobile phone cameras. Decimal numbers are rarely used as watermarks in watermarking algorithms. The technique was found to be very robust against attacks and produced the highest invisibility quality throughout this research. At the same time, the processing time of around 4 seconds is very high for an algorithm to be used inside mobile cameras or digital cameras.

- It has been found that the multiple embedding-extracting process is very useful for robustness, it can be concluded from the demonstrated results that a number of 5 times embedding is enough to maintain the desired robustness. At the same time when using a watermark size of 32×32 the number of multiple embedding and extracting processes is more than is required and wasteful of processing time, since the watermark will be embedded and extracted 32 times. The following table demonstrates the multiple embedding extraction times using different watermark sizes.

Table 6-4 Number of multiple embedding- extraction for each proposed algorithm

Number of multiple embedding-extraction processes		
<i>Algorithms</i>	<i>Watermark size</i>	<i>Number of embedding-extraction processes</i>
1,2,4,5	224×128	1
1,2,4,5	192×64	2
1,2,4,5	96×64	5
1,2,4,5	64×64	8
1,2,4,5	32×32	32
3,6	224×128	2
3,6	192×64	5
3,6	96×64	10
3,6	64×64	16
3,6	32×32	64

- Finally it can be concluded that the new algorithms outperform the current algorithms in the open literature. The proposed algorithm can achieves higher PSNR values after adjusting the watermarking strengths. The algorithms here are found to be robust against JPEG compression up to 15%, cropping up to 80%, small degrees of rotation up to 0.50 degree, scaling down up to 60%, additive noise, filtering operations and all Stirmark attacks except synchronization attacks.

6.4 Recommendations for Future Work

There are several directions of research that may broaden this work to gain more benefits and improve the performance. Recommendations for future work and suggestions can be introduced as follows:

- Most of the mobile phone and digital camera devices at the present time use a minimum of 2 mega pixels camera devices that can produce a high resolution images, much higher than the host images size (512×512) that were used in this research. An improvement in processing time when using high resolution images should be investigated. The processing time of the proposed algorithms might be chosen to be reduced in the future work by reducing the number of loops inside the code or reducing the number of embedding-extracting the watermark. For example the watermark of size 32×32 is embedded and extracted 32 times when using a host image of 512×512 . As concluded previously, a number of 5 times embedding is enough and can produce the desired results as shown in algorithms (1, 2, 4 and 5). When using a watermark size of 96×64 . This can be achieved by selecting some host sub-blocks for embedding. Use of the median rather than the mean when extracting the multiple embedded watermark and error correction codes in the embedding of watermarks could also be investigated.
- Watermarking strength is the scaling factor used to balance the quality against the robustness of the proposed algorithms. The watermarking strength must be chosen considering the changing nature and the content of the original images and the watermarks. An adaptive watermarking strength system must be opted to make the proposed algorithms more intelligent and to increase its adaptation capability toward the selection of the best watermarking strength.
- Colour watermarks may be tried instead of binary watermarks. Algorithm 6, with high capacity features, can be used to accommodate colour watermarks. For example if a colour watermark of size 32×32 is used, the total number of binary bits is equal to 24576. A colour watermark with this size can be embedded two times in a host image of size 512×512 .

- Real time implementation of the proposed schemes, especially algorithm 7, will have a great impact on the watermarking community. For example, Texas Instruments OMAP technology can be applied for real time implementation of the proposed algorithms. OMAP3 is used widely in industries in many mobile phone brands. OMAP3 is equipped with DSP and digital image processors.
- Fingerprinting is one prominent application and might be chosen to be tried as watermark in a DWT based watermarking system. Finger printing is used for identification proof in FBI data banks.
- The digital watermarking algorithms in this thesis should be tried on different applications, especially for audio and video applications. This is helpful in the watermarking of multimedia products. This feature is favourable for the implementation of audio and image/video watermarking algorithms on a common hardware.
- Medical image watermarking is still an open field of research. The developed watermarking schemes in this research can be useful techniques for medical images and must be tested in the future. For example, the proposed watermarks could be used for the authentication of medical images.
- Geometric distortions remain a challenging attack against many watermarking scheme and have to be chosen to be the major direction of future work. The algorithms proposed here can only survive small degrees of rotation up to 0.5 degree.

References

- [1] N. F. Johnson, "An introduction to watermark recovery from images," in *SANS Intrusion Detection and Response Conference (IDR '99)*, San Diego, CA, 1999., pp. 1-6.
- [2] S. Katzenbeisser and F. Petitcola, *Information Hiding Techniques for Steganography and Digital Watermarking*. London: Artech House., 2000.
- [3] I. J. Cox and M.L.Miller, "A review of watermarking and the importance of perceptual modeling," in *SPIE Electronic Imaging 97, storage and Retrieval for image and video Databases* San Jose, CA, 1997.
- [4] "Digital Watermarking Alliance " in <http://digitalwatermarkingalliance.org/>.
- [5] "Digimarc Corporation " in <https://www.digimarc.com/solutions/images/>.
- [6] G. C. Langelaar, I. Setyawan, and R. L. Lagendijk, " Watermarking Digital Image and Video Data," in *IEEE Signal processing magazine*, 2000.
- [7] A. M. Alattar, "Smart images using Digimarc's watermarking technology," in *SPIE Electronic Imaging '00, Security and watermarking of Multimedia content II*, San Jose, CA., 2000, pp. 264-273.
- [8] R. B. Wolfgang and E.J.Delp, "Fragile Watermarking using the VW2D watermark," in *Security and Watermarking of Multimedia Contents*, San Jose, CA, 1999, pp. 204-213.
- [9] M. Kutter and F. A. P. Petitcolas, "A Fair benchmark for image watermarking systems " in *Electronic Imaging '99. Security and Watermarking of Multimedia Contents, The international Society for optical Engineering*, , Sans Jose, USA, 1999.
- [10] R. Liu and T. Tan, "An SVD-Based Watermarking Scheme for Protecting Rightful Ownership," in *IEEE Transactions on Multimedia*, 2002, pp. 121 -128.
- [11] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Multi-resolution Scene-Based Video Watermarking using Perceptual Models," in *IEEE Journal on selected areas in communications*, 1998.
- [12] A. Abo-Zaid, "Robust and Invariant DFT Image Watermark: A New Approach Using Moments Invariant," in *International Conference on Information Technology*, 2002, pp. 130-137.
- [13] G. Voyatzis and I. Pitas, "Protecting Digital Image Copyrights: A Frame work," in *IEEE Computer Graphics and Applications*, 1999, pp. 18-24.
- [14] I. J. Cox, J. Kilian, F. Leighton, and T. Shamoan, "Secure Spread Spectrum Watermarking for Multimedia," in *IEEE Transactions on Image Processing*, 1997, pp. 1673 -1687.
- [15] S. Voloshynovskiy, S. Pereira, and T. Pun, "Attacks on Digital Watermarks: Classification, Estimation-Based Attacks, and Benchmarks", in *IEEE Communications Magazine*, 2001, pp. 118-26.
- [16] A. M. B. Sewaif, "Digital Image Watermarking Using Walsh Coded Handwritten Signatures," in *Department of Electronic Engineering*. vol. M.Sc. Sharjah: Etisalat University College, 2005, p. 111.
- [17] Shelby Pereira, S. Voloshynovskiy, M. Madueño, S. Marchand-Maillet, and T. Pun, "Second generation benchmarking and application oriented evaluation," in *In Information Hiding Workshop III*, Pittsburgh, PA, USA, April 2001.
- [18] V. Solachidis, A. Tefas, N. Nikolaidis, S. Tsekeridou, A. Nikolaidis, and I.Pitas, "A benchmarking protocol for watermarking methods," in *IEEE Int. Conf. on Image Processing (ICIP'01)*, Thessaloniki, Greece, October, 2001 pp. 1023-1026.
- [19] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Attacks on copyright marking systems," in *David Aucsmith (Ed), Information Hiding, Second International Workshop*, Portland, Oregon, U.S.A., 1998, pp. 219-239.
- [20] "The USC-SIPI Image Database," in <http://sipi.usc.edu/database/>.
- [21] M. Prasad and S. Koliwad, "A Comprehensive Survey of Contemporary Researches in Watermarking for Copyright Protection of Digital Images," in *International Journal of Computer Science and Network Security IJCSNS*, April 2009, pp. 91-107.
- [22] "Kodak Images," in <http://r0k.us/graphics/kodak/>.
- [23] B. M. Planitz and A. J. Maeder, "Medical Image Watermarking: A Study on Image Degradation," in *Proceedings of the Australian Pattern Recognition Society (APRS) Workshop on Digital Image Computing (WDIC 2005)*, Brisbane, Australia, 2005, pp. 3-8.

- [24] W.-N. Lie, G.-S. Lin, C.-L. Wu, and T. C. Wang, "Robust Image Watermarking on DCT Domain," in *ISCAS 2000 - IEEE International Symposium on circuits and Systems*, Geneva, Switzerland, 2000, pp. 228-231.
- [25] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," in *IEEE Transactions on Image Processing*, 2004, pp. 600-612.
- [26] Y. K. Lee and L. H. Chen, "High capacity image steganography model," in *IEE Proceedings - Vision, Images and Signal processing*, 2000, pp. 288-294.
- [27] R. v. Schyndel, A. Tirkel, and C. Osborne, "A Digital Watermark," in *proceedings of the 1st IEEE International Conference on Image Processing (ICIP)*, Austin, Texas, USA, , 1994, pp. 86-90.
- [28] A. Al-Jaber and I. Aloqily, "High Quality Steganography Model with Attacks Detection," *Pakistan Journal of Information And Technilogy*, vol. 2, pp. 116-127, 2003.
- [29] P.S.Huang, C. S. Chiang, C. P. Chang, and T. M. T. Vision, "Robust spatial watermarking technique for color images via direct saturation adjustment," in *IEE proceedings, Image and Signal Processing 2005*, pp. 561-574.
- [30] B. Verma, S. Jain, D. P. Agarwal, and A. Phadikar, "A new color image watermarking scheme," *Infocomp Journal of computer Science*, vol. 5, pp. 37-42, 2006.
- [31] I. nasir, Y. weng, and J. Jiang, "A new robust Watermarking Scheme for color Image in spatial domain," in *IEEE the third international conference on signal-image technology & internet-based systems (sitis' 2007)*, 2007.
- [32] A. A.-T. Al-Nu'aimi and R. Qahwaji, "An adaptive digital colored images watermarking technique using YCbCr," in *The 18th National Computer Conference, NCC*, KSA, 2006.
- [33] A. A.-T. Al-Nu'aimi and R. Qahwaji, "Adaptive watermarking for digital coloured images based on the energy of edges," in *IEEE International Conference on Signal Processing and Communications (ICSPC 2007)* Dubai, United Arab Emirates, 2007, pp. 1371-1374.
- [34] M. Barni, F. Bartolini, V. Cappellini, and A. Piva, "A DCT – domain system for robust image watermarking," in *Signal Processing, ELSEVIER*. vol. 66, 1998, pp. 357-372.
- [35] R. Hovancak and D. Levicky, "Digital image watermarking in different color models," in *2nd Slovakian – Hungarian Joint Symposium on Applied Machine Intelligence " SAMI 2004"* Herlany, Slovakia, 2004.
- [36] C.-S. Lu, S.-K. Huang, C.-J. Sze, and H.-Y. Liao, "Cocktail Watermarking for Digital Image Protection," in *IEEE Transactions on Multimedia*, 2000, pp. 209-224.
- [37] C.-H. Lee and Y.-K. Lee, "An Adaptive Digital Image Watermarking Technique for copyright protection," in *IEEE Transactions on Consumer Electronics*, 1999, pp. 1005-1015.
- [38] F. Deng and B. WangNanjing, "A Novel Technique for Robust Image Watermarking in the DCT Domain," in *IEEE International conference, Neural Networks & Signal Processing*, china, 2003.
- [39] Y. Y. CHUNG, "High capacity Digital watermarking system," in *proceeding of world academy of science, engineering and technology*, 2004.
- [40] J. Huang, Y. Q, and Y. Shi, "Embedding Image Watermarks in DC Components " in *IEEE Transactions on Circuts and systems for Video Technology*, 2000.
- [41] X.-M. Niu, Z.-M. Lu, and S.-H. Sun, "Digital watermarking of still image with gray-level digital watermarks," in *IEEE Transactions on Consumer Electronics*, 2000, pp. 137-145.
- [42] J. O. Ruanaidh and T. Pun, "Rotation, Scale and translation invariant spread spectrum digital image watermarking," in *Signal Processing, ELSEVIER*,, 1998, pp. 303-317.
- [43] C.-T. Hsu and J.-L. Wu, "Multiresolution Watermarking for Digital Images," in *IEEE Transactions on Circuits and Systems - II, Analog and Digital Signal Processing*, 1998, pp. 1097-1101.
- [44] D.-C. Lou and T.-L. Yin, "Adaptive Digital Watermarking Using Fuzzy Clustering Technique," in *IEICE Transactions on Fundamentals*, 2001, pp. 2052-2060.
- [45] C.-T. Hsieh and M.-Y. Hsieh, "A high robust blind watermarking algorithm in DCT domain," in *Proceeding of 9th international conference, KES 2005*,, Melbourne, Australia, 2005, pp. 1205-1211.
- [46] P. Meerwald and A. Uhl, "A Survey Of Wavelet-Domain Watermarking Algorithms," in *SPIE, Electronic Imaging, Security and Watermarking of Multimedia Contents III* 2001, pp. 505-516.
- [47] A. Al-Haj, "Combined DWT-DCT Digital Image Watermarking," in *Journal of Computer Science* 2007, pp. 740-746.

- [48] X. Cheng, L. Ding, and F. Gao, "Adaptive robust watermarking algorithm based on DCT-domain ", IEEE International Conference on Network Infrastructure and Digital Content, IC-NIDC 2009, pp. 623 - 628.
- [49] S. Sun, J. Ling, F. Dong, and J. Wan, "A New General Binary Image Watermarking in DCT Domain " in *International Seminar on Future BioMedical Information Engineering, FBIE* 2008, pp. 34 - 36.
- [50] D. Zhang, Z. Pan, and H. Li, " A novel watermarking algorithm in DCT domain to authenticate image content " in *IEEE International Conference on Intelligent Computing and Intelligent Systems, ICIS .* , 2009, pp. 608 - 611.
- [51] N.M.Charkari and M.A.Z.Chahooki, "A robust high capacity watermarking based on DCT and spread spectrum " in *IEEE International Symposium on Signal Processing and Information Technology*, 2007, pp. 194 - 197.
- [52] T. Zhang and Y. Du, " A digital watermarking algorithm for colour images based on DCT " in *International Conference on Information Engineering and Computer Science, ICIECS* 2009, pp. 1-4.
- [53] K. A. Ahmed, H. Al Ahmad, and P. Gaydecki, "A blind block based DCT watermarking technique for gray level images using one dimensional Walsh coding," in *International Conference on the Current Trends in Information Technology, CTIT*, 2009, pp. 1-6.
- [54] X. Gao, C. Qi, and H. Zhou, " An adaptive compressed DCT domain watermarking " in *8th International Conference on Signal Processing* 2006.
- [55] Z. Wei, J. Dai, and J. Li, " Genetic watermarking based on DCT domain techniques " in *Canadian Conference on Electrical and Computer Engineering, CCECE* 2006, pp. 2365 - 2368
- [56] Z. Rui-mei, L. Hua, P. Hua-wei, and H. Bo-ning, "A blind watermarking algorithm based on DCT " in *Second International Symposium on Intelligent Information Technology Application, IITA* 2008, pp. 821 - 824.
- [57] G. Zeng and Z. Qiu, "Image watermarking based on DC component in DCT " in *International Symposium on Intelligent Information Technology Application Workshops, IITAW* 2008, pp. 573 - 576.
- [58] Z. Yong, L. Li-Cai, L. Q. Shen, and J. Z. Tao, "A blind watermarking algorithm based on block DCT for dual colour images " in *Second International Symposium on Electronic Commerce and Security, ISECS* 2009, pp. 213 - 217.
- [59] Y. Zhou and J. Liu, " Blind watermarking algorithm based on DCT for colour images," in *2nd International Congress on Image and Signal Processing, CISP* 2009, pp. 1-3.
- [60] Z. Jiang-bin, Z. Yan-ning, F. Da-gan, and Z. Rong-chun, "Colour image watermarking based on DCT-domains of colour channels," in *IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering, TENCON*, 2002, pp. 281 - 284.
- [61] A. B. Sewaif, M. Al-Mualla, and H. Al-Ahmad, "Walsh-Coded Signatures for Robust Digital Image Watermarking," in *Proceedings of the 2004 IEEE Region 10 Conference - Analog and Digital Techniques in Electrical Engineering (TENCON2004)*, Chiang Mai, Thailand, 2004, pp. 431-434.
- [62] A. B. Sewaif, M. Al-Mualla, and H. Al-Ahmad, "2 D Walsh Coding for Robust Digital Image Watermarking," in *proceedings of the 4th IEEE International Symposium on Signal Processing and Information Technology (ISSPIT2004)*, Rome, Italy, 2004, pp. 302-305.
- [63] P.S.Huang, C. S. Chiang, C. P. Chang, and T. M. Tu, "Robust spatial watermarking technique for color images via direct saturation adjustment," in *Vision , Image and Signal Processing, IEE proceedings*, 2005, pp. 561-574.
- [64] B. Verma, S. Jain, D. P. Agarwal, and A. Phadikar, "A new color image watermarking scheme," *Infocomp, Journal of computer Science*, vol. 5, pp. 37-42, 2006.
- [65] S. Armeni, D. Christodoulakis, I. Kostopoulos, Y. Stamatou, and M. Xenos, "A Transparent Watermarking Method for Color Images," in *1st IEEE Balkan Conference On Signal Processing, Communications, Circuits, and Systems*, Maslak, Istanbul, Turkey, 2000.
- [66] W. Luo and G. L. Heileman, "A fast and robust watermarking method for JPEG images," in *Computer modeling & new Technologies*, 2004, pp. 39-47.
- [67] Y. Y. CHUNG, "High Capacity Digital Watermarking Systems," in *Proceedings Of World Academy Of Science, Engineering And Technology*, April 2004
- [68] M.Barni, F.Bartolini, A. D. Rosa, and A. Piva, "Capacity of the Watermark-Channel: How Many Bits Can Be Hidden Within a Digital Image?," in *Proc. of S P I E*, Jan 1999.

- [69] S. Shefali and S. M. Deshapande, "Mathematical Model for Improved Capacity Estimations for Data Hiding Techniques under Lossy Compression," in *Proceedings of the 2nd IMT-GT Regional Conference in Mathematics, Statistics and Applications*, Malaysia, June 2006.
- [70] X. Xu, M. Tomlinson, M. Ambroze, and M. Ahmed, "Techniques to Provide Robust and High Capacity Data Hiding of ID Badges for Increased Security Requirements," in *3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, SETIT 2005*, TUNISIA, 2005.
- [71] Y.J.Song, R.Z.Liu, and T.N.Tan, "Digital Watermarking for Forgery Detection in Printed Materials," in *International Symposium on Multimedia Information Processing*, Beijing, China, October 24-26, 2001.
- [72] J.-S. Sohn, S.-I. Lee, and D.-G. Kim, "Image adaptive watermarking technique for digital phone," in *International Conference on Computational Intelligence and Security*, Guangzhou, China, 2006, pp. 1190 - 1194.
- [73] A. Katayama, R. Kitahara, H. Kawamura, and H. Koike, "A Photo Verification Technique Using Embedded GPS Data for Mounting in Programmable Portable Phone Terminals," in *International conference in consumer electronics 2009*, p. 2.
- [74] S. B. Patel, "Proposed Secure Mechanism for Identification of Ownership of Undressed Photographs or Movies Captured using Camera Based Mobile Phones," in *Journal of Information Assurance and Security 2* 2007, p. 297-302.

Appendix A-Publications

PUBLICATIONS	
Paper1	A. Al-Gindy , H. Al-Ahmad, R. Qahwaji, and A. Tawfik, "Enhanced DCT based technique with shuffle scheme for robust image watermarking of handwritten signatures.,", in <i>proceeding of ICCCP'07 International Conference for Communication, Computer and Power</i> , Muscat, Oman, 2007, pp. 450-455
Paper2	A. Al-Gindy , H. Al-Ahmad, R. Qahwaji, and A. Tawfik, "A New Blind Image Watermarking of Handwritten Signatures Using Low-Frequency Band DCT Coefficients," in <i>In proceeding of ICSPC, the IEEE International Conference on Signal Processing and Communications</i> Dubai, United Arab Emirates, 2007, pp. 1367-1370.
Paper3	A. Al-Gindy , H. Al-Ahmad, R. Qahwaj, and A. Tawfik, "New blind Image watermarking technique for dual watermarks using low-frequency band DCT coefficients," in <i>In proceeding of ICECS the 14th IEEE International Conference on Electronics, Circuits and Systems</i> , Marrakech, Morocco., 2007, pp. 538-541.
Paper4	A. Al-Gindy , H. Al-Ahmad, R. Qahwaj, and A. Tawfik, "A novel blind Image watermarking technique for colour RGB images in the DCT domain using green channel " in <i>Mosharaka International Conference on Communications, Computers and Applications (MIC-CCA 2008)</i> , , Amman, Jordan, 2008, pp. 26-31.
Paper5	A. Al-Gindy , H. Al-Ahmad, R. Qahwaj, and A. Tawfik, "Watermarking of colour images in the DCT domain using Y channel," in <i>IEEE/ACS International Conference on Computer Systems and Applications</i> , Rabat, Morocco, 2009, pp. 1025-1028.
Paper6	A. Al-Gindy , H. Al-Ahmad, R. Qahwaj, and A. Tawfik, "A new watermarking scheme for colour images captured by mobile phone cameras," in <i>The Ninth IASTED International Conference on Visualization, Imaging and Image Processing</i> , Cambridge, UK, 2009.
Paper7	A. Al-Gindy , H. Al-Ahmad, R. Qahwaj, and A. Tawfik, "A frequency domain adaptive watermarking algorithm for still colour images," in <i>International Conference on Advances in Computational Tools for Engineering Applications, ACTEA '09.</i> , Beirut, Lebanon, 2009, pp. 186-191.
Paper8	A. Al-Gindy , H. Al-Ahmad, R. Qahwaj, and A. Tawfik, "A new watermarking scheme for colour images captured by mobile phone cameras," <i>IJCSNS International Journal of Computer Science and Network Security</i> , vol. 9, pp. 248-254, July 2009.
Paper 9	A. Al-Gindy , H. Al-Ahmad, R. Qahwaj, and A. Tawfik, "A High Capacity Digital Watermarking Technique for the Authentication of Colour Images," in <i>9th IEEE International Symposium on Signal Processing and Information Technology</i> , Ajman, United Arab Emirates, 2009, pp. 37-42.