

University of Southern Queensland  
Faculty of Engineering & Surveying

**Watermarking for Securing Digital Media**  
**Content**

A dissertation submitted by

LOH Shiau Ping

in fulfilment of the requirements of

**ENG4112 Research Project**

towards the degree of

**Bachelor of Electrical and Electronic Engineering**

Submitted: October, 2004

# Abstract

With the proliferation of digitized media, the need of digital watermarks as a copyright protection, ownership identification and a secure way of embedding information has become important. A useful watermark technique should be robust against malicious attacks or tampering to remove the watermark and should not greatly affect the quality of the original file.

In conventional cryptographic systems, once the information is decrypted, the recipient can misuse it. The reproduction and retransmission cannot be tracked easily.

In these project, LSB watermarking technique for digital image is investigated. A software system, consisting of watermark embedding and recovery is implemented with single or multiple watermarks embedding. The robustness and effectiveness of this watermarking technique is tested using the GIF and JPEG file compression format.

A comparator is designed and implemented to further enhanced the robustness when subjected to compression. All the results of the LSB watermarking techniques are tabulated.

University of Southern Queensland  
Faculty of Engineering and Surveying

<b>ENG4111/2 <i>Research Project</i></b>
--

### **Limitations of Use**

The Council of the University of Southern Queensland, its Faculty of Engineering and Surveying, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Engineering and Surveying or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course pair entitled “Research Project” is to contribute to the overall education within the student’s chosen degree program. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

**Prof G Baker**

Dean

Faculty of Engineering and Surveying

# Certification of Dissertation

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

LOH SHIAU PING

0031135262

---

Signature

---

Date

# Acknowledgments

I would like to thank my Supervisor Dr. John Leis for his support, ideas, genuine interest and time during the duration of the whole project.

Most importantly I would like to thank my parents, brother, sister and friends for their constant ideas, support and motivation.

LOH SHIAU PING

*University of Southern Queensland*

*October 2004*

# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgments</b>	<b>iv</b>
<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xvii</b>
<b>Chapter 1 Introduction</b>	<b>1</b>
1.1 Aims and Objectives . . . . .	1
1.2 Methodology . . . . .	1
1.3 Scope and Limitations of the Research . . . . .	3
1.4 Research Approach . . . . .	4
1.5 Outline of Dissertation . . . . .	4
<b>Chapter 2 Digital Management Rights</b>	<b>6</b>
2.1 Digital Management Rights . . . . .	6

<b>CONTENTS</b>	<b>vi</b>
2.2 DRM Systems Functionality . . . . .	8
2.2.1 Packaging Rules Generation and Modifications . . . . .	8
2.2.2 Value Chain Management and License Services . . . . .	9
2.2.3 Consumption Services . . . . .	10
2.2.4 Trust Management Services . . . . .	11
2.2.5 Security and Protected Platform Services . . . . .	11
2.3 Benchmarking of Robust watermarking for DRM . . . . .	12
2.3.1 Digital Cinema . . . . .	13
2.3.2 Broadcasting of Images . . . . .	13
2.3.3 Contribution Links . . . . .	13
2.3.4 Internet Distribution of Images . . . . .	14
2.4 Legal Policy and Digital Rights Management . . . . .	14
2.4.1 DRM and Copyright Law . . . . .	15
2.4.2 DRM and Contract . . . . .	15
2.4.3 DRM and Privacy . . . . .	15
2.4.4 DRM and Competition Law . . . . .	16
<b>Chapter 3 Current Research in Watermarking</b>	<b>17</b>
3.1 Information Hiding Techniques . . . . .	17

<b>CONTENTS</b>	<b>vii</b>
3.1.1 Encryption . . . . .	18
3.1.2 Cryptography . . . . .	18
3.1.3 Steganography . . . . .	20
3.2 Digital Watermarking for Still Images . . . . .	23
3.2.1 Introduction . . . . .	23
3.2.2 Properties of Watermarks . . . . .	24
3.2.3 Types of Watermarks . . . . .	26
3.2.4 Applications of Watermarks . . . . .	27
3.2.5 Watermarking Techniques . . . . .	30
3.3 Digital Watermarking for Audio . . . . .	32
3.3.1 Introduction . . . . .	32
3.3.2 Properties of Audio Watermarks . . . . .	33
3.3.3 Applications of Audio Watermarks . . . . .	34
3.4 Attacks and Benchmarks of Digital Watermarking Systems . . . . .	36
3.4.1 Threats and Risks . . . . .	36
3.4.2 Classifications of Attacks . . . . .	37
3.4.3 Benchmarking . . . . .	40



<b>CONTENTS</b>	<b>viii</b>
4.1 Algorithm . . . . .	43
4.2 Test Images . . . . .	44
4.3 Software . . . . .	48
4.3.1 MATLAB . . . . .	48
4.3.2 IrfanView . . . . .	48
4.4 Program . . . . .	49
4.4.1 MATLAB code . . . . .	49
4.4.2 Flowchart of Source Code . . . . .	56
4.4.3 Graphical User Interface (GUI) . . . . .	63
4.5 Results . . . . .	65
4.5.1 Joint Photographic Experts Group (JPEG) . . . . .	65
4.5.2 Graphic Interchange Format (GIF) . . . . .	77
4.5.3 Improving the Basic Algorithm . . . . .	79
<b>Chapter 5 Conclusion and Future Work</b>	<b>80</b>
5.1 Achievement of Project Objectives . . . . .	81
5.2 Further Work . . . . .	82
<b>References</b>	<b>84</b>
<b>Appendix A Project Specification</b>	<b>87</b>

<b>CONTENTS</b>	<b>ix</b>
<b>Appendix B Detail Test Results</b>	<b>89</b>
B.1 Single Watermark . . . . .	90
B.1.1 Bird . . . . .	90
B.1.2 Lena . . . . .	92
B.1.3 Clock . . . . .	95
B.1.4 Bridge . . . . .	97
B.1.5 Camera . . . . .	100
B.2 Multiple Watermarks without Comparator . . . . .	102
B.2.1 Bird . . . . .	102
B.2.2 Lena . . . . .	105
B.2.3 Clock . . . . .	107
B.2.4 Bridge . . . . .	110
B.2.5 Camera . . . . .	112
B.3 Multiple Watermark with Comparator . . . . .	115
B.3.1 Bird . . . . .	115
B.3.2 Lena . . . . .	117
B.3.3 Clock . . . . .	120
B.3.4 Bridge . . . . .	122
B.3.5 Camera . . . . .	125

---

**Appendix C Source Code** **128**

C.1	Main.m . . . . .	129
C.2	Main1.m . . . . .	130
C.3	Main2.m . . . . .	131
C.4	Main3.m . . . . .	132
C.5	Main4.m . . . . .	134
C.6	Main5.m . . . . .	134
C.7	Single_embed.m . . . . .	135
C.8	Multiple_embed.m . . . . .	137
C.9	Single_decode.m . . . . .	139
C.10	With_comparator.m . . . . .	140
C.11	Without_comparator.m . . . . .	142

# List of Figures

2.1	DRM System basic reference model . . . . .	8
3.1	Illustration of a Cryptographic System . . . . .	19
3.2	Illustration of a Steganographic System . . . . .	21
4.1	Test image : bird.bmp (256x256 pixels) . . . . .	44
4.2	Test image : camera.bmp (256x256 pixels) . . . . .	45
4.3	Test image : lena.bmp (256x256 pixels) . . . . .	45
4.4	Test image : clock.bmp (256x256 pixels) . . . . .	46
4.5	Test image : bridge.bmp (256x256 pixels) . . . . .	47
4.6	Test image : copyright.bmp (12x9 pixels) . . . . .	47
4.7	Main.m : LSB watermarking main menu) . . . . .	49
4.8	Select image menu . . . . .	50
4.9	Select watermarking scheme menu . . . . .	51
4.10	Select JPEG compression quality menu . . . . .	52

## LIST OF FIGURES xii

---

4.11 Single embed watermark retrieval . . . . .	52
4.12 Multiple embed watermark retrieval . . . . .	53
4.13 Flowchart of overall program . . . . .	57
4.14 Flowchart of overall program . . . . .	58
4.15 Flowchart of single watermark embedding . . . . .	59
4.16 Flowchart of single watermark embedding . . . . .	60
4.17 Flowchart of single watermark decoding . . . . .	61
4.18 Flowchart of multiple watermark decoding without comparator . .	62
4.19 Flowchart of multiple watermark decoding without comparator . .	63
4.20 Graphical User Interface (GUI) . . . . .	64
4.21 Test results for Bird.bmp . . . . .	67
4.22 Test results of Lena.bmp . . . . .	69
4.23 Test results of Clock.bmp . . . . .	71
4.24 Test results of Bridge.bmp . . . . .	73
4.25 Test results of Camera.bmp . . . . .	75
 B.1 Test result with 100% compression quality factor . . . . .	 90
B.2 Test result with 99% compression quality factor . . . . .	90
B.3 Test result with 98% compression quality factor . . . . .	91

**LIST OF FIGURES****xiii**

---

B.4	Test result with 95% compression quality factor . . . . .	91
B.5	Test result with 90% compression quality factor . . . . .	92
B.6	Test result with 100% compression quality factor . . . . .	92
B.7	Test result with 99% compression quality factor . . . . .	93
B.8	Test result with 98% compression quality factor . . . . .	93
B.9	Test result with 95% compression quality factor . . . . .	94
B.10	Test result with 90% compression quality factor . . . . .	94
B.11	Test result with 100% compression quality factor . . . . .	95
B.12	Test result with 99% compression quality factor . . . . .	95
B.13	Test result with 98% compression quality factor . . . . .	96
B.14	Test result with 95% compression quality factor . . . . .	96
B.15	Test result with 90% compression quality factor . . . . .	97
B.16	Test result with 100% compression quality factor . . . . .	97
B.17	Test result with 99% compression quality factor . . . . .	98
B.18	Test result with 98% compression quality factor . . . . .	98
B.19	Test result with 95% compression quality factor . . . . .	99
B.20	Test result with 90% compression quality factor . . . . .	99
B.21	Test result with 100% compression quality factor . . . . .	100
B.22	Test result with 99% compression quality factor . . . . .	100

---

B.23 Test result with 98% compression quality factor . . . . .	101
B.24 Test result with 95% compression quality factor . . . . .	101
B.25 Test result with 90% compression quality factor . . . . .	102
B.26 Test result with 100% compression quality factor . . . . .	102
B.27 Test result with 99% compression quality factor . . . . .	103
B.28 Test result with 98% compression quality factor . . . . .	103
B.29 Test result with 95% compression quality factor . . . . .	104
B.30 Test result with 90% compression quality factor . . . . .	104
B.31 Test result with 100% compression quality factor . . . . .	105
B.32 Test result with 99% compression quality factor . . . . .	105
B.33 Test result with 98% compression quality factor . . . . .	106
B.34 Test result with 95% compression quality factor . . . . .	106
B.35 Test result with 90% compression quality factor . . . . .	107
B.36 Test result with 100% compression quality factor . . . . .	107
B.37 Test result with 99% compression quality factor . . . . .	108
B.38 Test result with 98% compression quality factor . . . . .	108
B.39 Test result with 95% compression quality factor . . . . .	109
B.40 Test result with 90% compression quality factor . . . . .	109
B.41 Test result with 100% compression quality factor . . . . .	110

---

B.42 Test result with 99% compression quality factor . . . . .	110
B.43 Test result with 98% compression quality factor . . . . .	111
B.44 Test result with 95% compression quality factor . . . . .	111
B.45 Test result with 90% compression quality factor . . . . .	112
B.46 Test result with 100% compression quality factor . . . . .	112
B.47 Test result with 99% compression quality factor . . . . .	113
B.48 Test result with 98% compression quality factor . . . . .	113
B.49 Test result with 95% compression quality factor . . . . .	114
B.50 Test result with 90% compression quality factor . . . . .	114
B.51 Test result with 100% compression quality factor . . . . .	115
B.52 Test result with 99% compression quality factor . . . . .	115
B.53 Test result with 98% compression quality factor . . . . .	116
B.54 Test result with 95% compression quality factor . . . . .	116
B.55 Test result with 90% compression quality factor . . . . .	117
B.56 Test result with 100% compression quality factor . . . . .	117
B.57 Test result with 99% compression quality factor . . . . .	118
B.58 Test result with 98% compression quality factor . . . . .	118
B.59 Test result with 95% compression quality factor . . . . .	119
B.60 Test result with 90% compression quality factor . . . . .	119



## LIST OF FIGURES

xvi

---

B.61 Test result with 100% compression quality factor . . . . .	120
B.62 Test result with 99% compression quality factor . . . . .	120
B.63 Test result with 98% compression quality factor . . . . .	121
B.64 Test result with 95% compression quality factor . . . . .	121
B.65 Test result with 90% compression quality factor . . . . .	122
B.66 Test result with 100% compression quality factor . . . . .	122
B.67 Test result with 99% compression quality factor . . . . .	123
B.68 Test result with 98% compression quality factor . . . . .	123
B.69 Test result with 95% compression quality factor . . . . .	124
B.70 Test result with 90% compression quality factor . . . . .	124
B.71 Test result with 100% compression quality factor . . . . .	125
B.72 Test result with 99% compression quality factor . . . . .	125
B.73 Test result with 98% compression quality factor . . . . .	126
B.74 Test result with 95% compression quality factor . . . . .	126
B.75 Test result with 90% compression quality factor . . . . .	127

# List of Tables

3.1	Table of Attacks . . . . .	38
4.1	JPEG compression quality table for single and multiple watermarks	76
4.2	GIF compression quality table for single and multiple watermarks	78

# Chapter 1

## Introduction

### 1.1 Aims and Objectives

The aim of this project is to experimentally evaluate the effect of compression on embedded watermarks in digital media. Current research in the area of image and audio watermarking is to be investigated, and the robustness of simple watermarking methods to lossy compression should be experimentally evaluated using a suitable software platform.

The algorithm is designed based on the watermarking proposed in the literature and coded using MATLAB software. In addition, their effectiveness will be determined when subjected to compression.

### 1.2 Methodology

- Research the background information relating to watermarking and other information hiding techniques.

This is accomplished by researching for relevant materials from the libraries

---

and Internet.

- Research the possible application areas of digital watermarking.

After gaining knowledge of the background information relating to watermarking, the application areas are studied. This allows a better understanding on how the watermarks are used in different applications. Most of the materials are found in libraries and online materials

- Investigate several different watermarking algorithms.

The different watermarking techniques are studied before the actual designing and programming of the algorithm. The types watermarking algorithms are research from the background of information hiding techniques and digital watermarking.

To ensure that the algorithms designed are accurate, software codes are written and programmed to verify that data can be watermarked. Therefore the software that is used to program the watermarking algorithms is chosen and able to handle the file format. Codes and build-in routines used are familiarized before the coding of the designed watermarking algorithm.

- Implement one or more watermarking techniques and experimentally investigate the ability to recover the watermark when subjected to compression/decompression using JPEG and GIF encoding.

The LSB watermarking technique is implemented in this project after gaining the knowledge relating to watermarking. After the LSB watermarking technique is designed and coded, the effectiveness of the watermarked image is tested using the JPEG and GIF format. This allowed the investigation of the LSB watermarking technique and the effectiveness of the watermark when retrieved after the compression.

- Investigate methods to improve the robustness of the watermark recovery when the image is subjected to lossy compression.

The method to further improve the LSB watermarking technique is by embedding multiple watermarks into the image at the same time. This allows the higher chances of recovering the watermark after subjected to compression. A comparator can be used to determine the final pixels value of the watermark retrieve which will result in a similar or more comparable to the original watermark image.

### **1.3 Scope and Limitations of the Research**

This is a research and software based project. The purpose is to implement a watermarking technique and test its effectiveness when subject to different compression techniques.

Materials from the Internet can be easily obtained. The books relating to the project are borrowed from the national library locally. The varieties of books are very limited and most of them can only be referenced in the library. Books from the university are more comprehensive but can only be borrowed by the students. I managed to get borrow some books through a friend studying there.

The main concern is when the books are on loaned to others or cannot be located. These factors will cause some of the project specifications that needed researching to be affected. Other resources to obtain the materials need to be sought.

## 1.4 Research Approach

The project specification is done prior to the project research. This will help to determine a more focus research area. Most of the materials research are found online on the Internet and in the local national libraries. These materials help to provide a background information and knowledge before the watermarking technique is decided.

When a certain amount of understanding is gained, the LSB watermarking marking is chosen to be implemented. Information relating to the LSB watermarking technique and related materials are intensively research. After the specific technique understanding is gained, the design and implementation of the LSB algorithm is done.

## 1.5 Outline of Dissertation

This dissertation is organised as follows:

**Chapter 2** describes on the Digital Management Rights (DRM), its different systems functionality, benchmarking, and legal policy and rights management.

**Chapter 3** describes the information hiding techniques, digital watermarking for different digital media content, and the attacks and benchmark of digital watermarking systems.

**Chapter 4** describes the test results of the LSB watermarking technique. The effectiveness and robustness of the watermarking technique is particularly studied.

**Chapter 5** describes the conclusion of the project and the future work that is possible to further enhance the project.

# Chapter 2

## Digital Management Rights

### 2.1 Digital Management Rights

Digital media distribution has been strongly pushed by the modern advancements in communication infrastructure, signal processing and digital storage technologies. Digital distribution allowed the introduction to flexible, cost-effective business models that are advantage to multimedia commerce transactions. The digital nature of the information enable individual to manipulate, duplicate, or access media beyond the conditions agreed upon for a given transaction.

Digital rights management (DRM) has been proposed to manage the digital management of user rights to content. Ideally, a DRM system balances information protection, usability and cost to provide a beneficial environment for all parties. It achieved the overall management through the interaction of effective economics models, social values, legal policy and technology. DRM also associates specific user rights to media in order to provide constant governance of user activities such as viewing, duplication and access.



With the fast advancement in technology, DRM systems incorporate encryption, copy control, tagging, tracing, conditional access and media identification. The challenge is to engineer secure systems in an environment of dynamic applications and standards for which appropriate business models and consumer expectations are now being identified.

DRM enable technically enforced licensing of the digital information. This allowed commercial publishers to be able to distribute valuable content electronically, without destroying the copyright holder's revenue stream. Therefore a well designed DRM system should provide the following:

- Governance

DRM is different from classical security and protection technologies. DRM implement, control, or governance, via the use of programming language methods executed in a secure environment.

- Secure Association of Usage Rules With Information

DRM systems securely associate rules with content. These rules determine the usage of the content throughout. Rules can be attached to content, embedded within content or can be delivered independently of content.

- Persistent Protection

DRM systems are designed to protect and govern information on a persistent basis throughout the content's commercial life cycle. Protection is frequently provided using cryptographic techniques. Encrypted content is protected even as it travels outside of protected distribution channels.

## 2.2 DRM Systems Functionality

The proposed basic DRM reference model is illustrated in Figure 2.1. The functional characteristics of the five main domains are explained as follows:

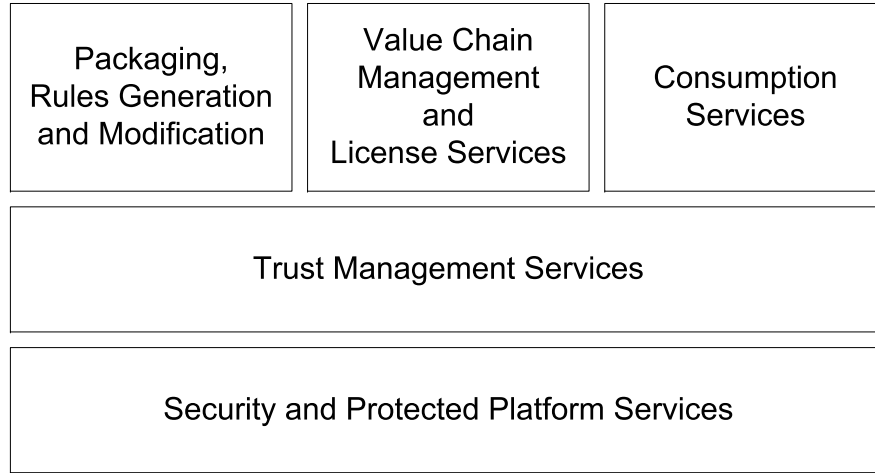


Figure 2.1: DRM System basic reference model

### 2.2.1 Packaging Rules Generation and Modifications

The point of entry to the DRM-managed content and governance life cycle includes technologies supporting content packaging, specification of rights and associated data, and generation and modification of digital items.

- Content Packaging

Content packaging is the process of preparing content for DRM protection usually by encrypting it, associating the necessary identifiers, logging and cataloging the content. Content identifiers couple the protected content with rules and content protection keys. Therefore the rules, packaged content and content keys may be generated together or separately at the same time or different times.

Content protection is accomplished using cryptographic processing where content protection keys are made available to one value chain participant and not revealed to other value chain participants.

- Rules Generation and Modification

Any authorized member of the value chain can create rules associated with a content package. The rules are used to govern consumer access to content and the information associated with the content. In some system, it is possible to modify or extend rules after their initial creation. The value chain management and licensing services may support the ability to select and apply rules that have been updated regardless when the content was packaged and placed into the system. The rules are then embedded into data structures that can be linked to the content.

### **2.2.2 Value Chain Management and License Services**

Consumer licenses are sometimes the result of a collaboration of multiple value chain participants. Authorized members may insert new rules into the licensing structures using the processes that are governed by them. Value chain management includes processing rules in the license associated with the content or creates as an electronic contract covering specific offers or content and delivered separately.

- Value Chain Management

Static value chain management refers to approaches where offer and consumption rules are computed at content packaging time.

This management is parameterized at packaging time with information about the known and identified participants and the packager output conveys the necessary information in advance of actual participation.

Dynamic value chain management, the rules governing the use of value chain information is accessed on demand through network services. The content are distributed by reference rather than copying packaged file to each value chain participant. The rights of the content are distributed based on these references and may be incorporated in licenses.

- **Licensing Processes**

DRM functions are closely associated with license services including the management of data structures carrying rules and cryptographic information. It also includes

- Discovery, delivery, authentication and management of offers.
- Validation of trusted status of entities requesting services of the system.
- Validation of transaction from peer value chain system authorizing generation and association of licenses on behalf of a third party.
- Management and enforcement of subscription data.
- Event reporting for payment, usage tracking and overall system assurance.

### **2.2.3 Consumption Services**

Consumption services are function through which consumers interact with DRM content according to some governed action (e.g. playback, editing, printing, etc.).

They are normally associated with consumer client systems but may also associated with any value chain participant that accesses or processes protected content or rules.

- **Consumption and Portable Devices**

Portable devices are another class of consuming systems. A host that is capable of direct transactions with distributed value chain management and licenses services usually manages the portable device. Portable devices rely on a secure communication channel managed by the host system for functions such as copying and re-associating protected content to the portable device for offline usage or rendering.

### **2.2.4 Trust Management Services**

Trust management services are responsible for functions supporting provisioning, certification, secure operations and renewability of elements in the distributed DRM systems. Trust management services are relied upon by features in virtually all components of the DRM system. Its management subsystems use authorization techniques to regulate activities with risk potential within and between DRM systems components.

### **2.2.5 Security and Protected Platform Services**

A trusted environment for persistent governance of rules and content is built on a foundation of security functions. The required security functions may control trusted hardware if it is available.

Security and protected platform and technologies include software tamper resistance whereby the host and device software and firmware is designed to provide protection of content buffers, persistent state, and key stores. The execution environment security allows the host and device to be validated with various integrity checks to ensure that it is a legitimate and has not been modified.

## **2.3 Benchmarking of Robust watermarking for DRM**

Digital rights management (DRM) systems are built from several components that allow setting efficient electronic commerce of intangible goods. A DRM system has to compromise between the security threats of the content owners, the privacy of the end user and the cost of the components that will be used to establish trust between parties.

In multimedia, the digital content has to be provided in an analog form at the end point, which can be easily captured and re-digitized for illegal redistribution. Therefore digital insertion of marks to individualize, trace and control usage of a digital Work, even when it is transformed into analog signals, will be one of the pillars of future DRM systems.

The aim of DRM is to analyze the potential security weakness in the distribution chain and identify at each point of the chain what tools have to be implemented as countermeasure. Some scenarios are address related to image distribution.

---

**2.3.1 Digital Cinema**

The content of the digital cinema distribution is exhibited in a theater room. Therefore the watermarking allows tracing the room identification and time of a projection, which should be rescanned by a camera during the exhibition. In this case, the retrieval of the parameters of an unauthorized copy can be done using the original version of the content. The digital content duplication for theaters consists of direct bit-to-bit copies done in the storage device. With the proper use of encryption and reliable key distribution, the illegal copying of digital content can be prevented.

**2.3.2 Broadcasting of Images**

In broadcasting, a specific content is broadcast to setup decoders, which the tracing of content and copy control can be done by watermarking. Content provider over broadcast channels are wary of any breach of contract whereby the content is shown more often or at other channels that has been agreed upon. Therefore monitoring stations are setup to verify the proper showing of the contents. In order to reduce the complexity and security issues of a monitoring station, content is being stamped with an invisible watermark that cannot easily be retrieved from the content after distribution. Copy control is achieved by using a control bit, which is tied to the content. This copy bit is implemented as a robust watermark.

**2.3.3 Contribution Links**

The contributions links are the liaison between content providers and studios. The providers are multicasting content, which are remastered at each studio to be distributed in secondary links.

The main DRM concern is to identify the copyright owner of the content when it has gone through several postproduction processes. Therefore the use of watermark containing the content owner's identity is a good solution to prove to a legal authority the ownership of a Work. The watermark is a good means to solve conflicts because it is very robust and not easily removable as the inscription is made with the use of a secret key. Only the owner of this key will be able to read or detect the watermark.

### **2.3.4 Internet Distribution of Images**

The alternative to copy control and trusted computers relies on the responsibilities of the content user and tools to mark this responsibility. Legal actions against copyright infringements on the Web have already decreases the amount of peer-to-peer redistribution of content. Therefore with watermark implementation, combined with registration authorities and transaction certifications help to improve user awareness in these issues.

## **2.4 Legal Policy and Digital Rights Management**

DRM is generally taken to refer to systems for describing and enforcing copyright associated with networked digital data distribution. With the proper design and implementation, DRM technology can enable an electronic market and maximize the utility of digital works for the whole community. A DRM system is a multiple systems compete and often rely on open standards since they are deployed in mass market. Therefore DRM laws are designed to respond to the advance of ingenious hackers.



---

**2.4.1 DRM and Copyright Law**

Copyright law is the essential underpinning of DRM. The development of this law is continuously spurred by the appearance of new copying technologies. There are no formal requirements for copyright but it arises automatically with the creation of a Work. Copyright confers to the copyright holder the rights of reproduction, communication and distribution of a creative work as well. This allows the author to reproduce a work, recover the investment made in the creative process and to profit from the outcome of the process. Copyright infringement occurs when one of these rights is breached and remedies to these include civil action and criminal charges.

**2.4.2 DRM and Contract**

Contracts are another source of legal rights permitting copyright owners to protect their intellectual property. In the mass-market goods, the contract consists of the notice of terms to which an acquirer must adhere as a condition of acquiring access to the good and no negotiation is allowed. The enforcement of such contracts is often limited and permitted to use the intellectual property on terms specified by the owner.

**2.4.3 DRM and Privacy**

In addition to concerns regarding copyright, there is also considerable unease about the effect that DRM will have on user privacy. DRM clients can be configured to collect usage data each time the user accesses content that caused serious intrusion into the privacy of the users. Privacy protection is implemented in a multitude of activities and contexts.

For this reason, the legal protection of privacy is protected by a patchwork of laws. However, privacy is by no means an absolute right and is only used when necessary to facilitate transactions

#### **2.4.4 DRM and Competition Law**

The aim of competition law is to encourage and protect competition in consumer and business-to-business markets for goods and services. This is because competitive activity promotes economic efficiency, resulting in lower services, better products and enhanced consumer welfare. Competition law seeks to preserve the competitive structure of markets by preventing situations in which market power undermines competition without offering a counterbalancing economic benefit.

# Chapter 3

## Current Research in Watermarking

### 3.1 Information Hiding Techniques

With the fast advancement in technology, the society has entered a new era which commercial activities, business transactions and government activities are conducted and offered over open computer and communication networks such as the Internet. This allowed the easy accessibility to anyone around world.

These services can only be allowed over the open networks if conducted in a secure manner. Therefore to provide an effective solution, information hiding techniques are used to secure communications over open networks. Encryption, digital signatures, password-based user authentication are some of the most common techniques for securing communications.

With the increasing demand for protecting communications over open networks and more sophisticated forms of electronic commerce, business and services

requires constant improvement in the security. A huge effort is needed from professional to design, developed, analyzed and maintained the information hiding techniques in order to allowed the updated solutions to counter the dangers faced by anyone in the open computer and communication networks. Some information hiding techniques are explained in the following sections.

### **3.1.1 Encryption**

Encryption is a process to transform a piece of information into an incomprehensible form. It is a practical means to achieve information secrecy. The input to the transformation is called plaintext and the output is called ciphertext.

In order to restore the information, an encryption transformation must be reversible and is called decryption. Encryption and decryption are used by cryptographic keys. An encryption algorithm, decryption algorithm, format of the messages and keys will form a cryptographic system.

Encryption is also a basis for algorithms used in steganography. The algorithms take a block of data and hide it in the noise of an image or sound file that is as close to random as possible. This lowers the chance that it can be detected.

### **3.1.2 Cryptography**

Cryptography can be defined as the processing of information into an unintelligible (encrypted) form for the purposes of secure transmission. It provides a means for secure communications. The receiver can only decode the encrypted message by the use of a 'key' to retrieve the original message. More advanced crypto techniques ensure that the information being transmitted has not been modified in transit. The cryptographic system is shown in Figure 3.1

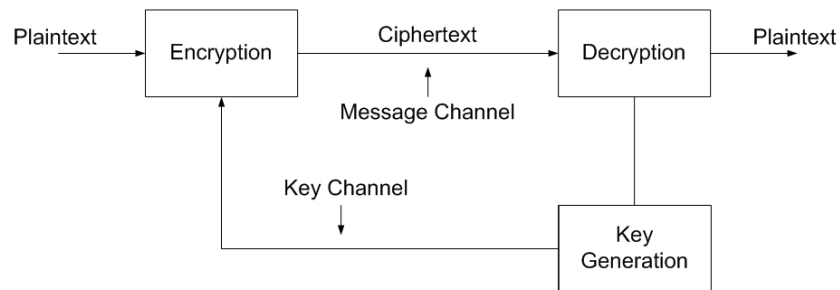


Figure 3.1: Illustration of a Cryptographic System

Cryptographic techniques generally rely on the metaphor of a piece of information being placed in a secure "box" and locked with a "key". The information itself is not disturbed and anyone with the proper key can gain access. Once the box is open, all of the information security is lost.

Cryptography consists of 3 types of encryption schemes: symmetric, asymmetric and hash (Cole 2003).

- Symmetric

Symmetric key encryption is a single-key encryption. It uses one key to encrypt the plaintext and the same key to decrypt the ciphertext. This encryption technique is straightforward and fast. The drawback of this technique is that the decryption cannot be done if the key is not sent over the secure channel.

- Asymmetric

Asymmetric encryption uses a public and private key. The plaintext is encrypted with the public key and the ciphertext is decrypted with the private key. The key has to be sent over a trusted channel to ensure that there is no modification done during the transit. Therefore the public key can be given to anyone who needs to encrypt the plaintext.

- Hash

A hash algorithm is a one-way transformation of the plaintext message that cannot be reversed. It takes the plaintext in any size and produces a smaller fixed-length output that is irreversible. Hash is useful for storing passwords and for digital signatures because there is no key.

A user password can be run in the hash algorithm. When a user log on to a system, the user will be prompted to enter a password. The password is then run through the hash algorithm and compared to the encrypted text. If they match, the user is granted access. If not, the access is denied.

A digital signature is added to a document with the sender's private key. Hash takes a message and produces a smaller, fixed-length output and then encrypts it with the sender's private key. Therefore the less information that has to be encrypted will make the process faster.

### 3.1.3 Steganography

Steganography is a term derived from the Greek word *steganos*, which means "covered writing". It improved on cryptography by hiding that a communication has occurred. Steganography is the study of techniques for hiding the existence of a secondary message in the presence of a primary message. The primary message is referred to as the *carrier message*; the secondary message is referred as the *payload message*. The Steganographic system is shown in Figure 4.6.

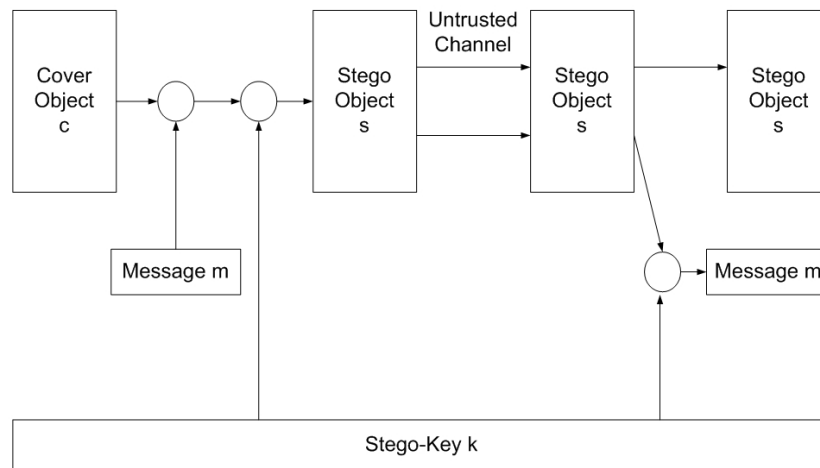


Figure 3.2: Illustration of a Steganographic System

The message  $m$  is imbedded into a harmless message  $c$ , which is defined as the cover-object. The message  $m$  is then embedded into  $c$ , generally with use of a key that is defined as the *stego-key*. The resulting message is then embedded into the cover-object  $c$ , which results in stego-object  $s$ . Ideally the *stego-object* is indistinguishable from the original message  $c$ , appearing as if no other information has been encoded (Katzenbeisser 1999). The hope of the system is that the stego-object will be close enough in appearance and statistics to the original such that the presence of information will be undetected.

Steganography can also be used to place a hidden 'trademark' in images, music and software, a technique referred to as watermarking.

Steganography is an effective method of hiding data in multimedia (e.g. image, audio and video) that has been used throughout history. Classical steganography systems depend on keeping the encoding system secret, but modern steganography is detectable only if secret information is known.

By using files that contain unused or insignificant areas of data and replacing them with the information, it hides a message within a larger one that appears to be part of the original file in such a way that others cannot discern the presence or contents of the hidden message (Provos 1999). This technique makes it impossible to detect anything inside the file, and only the intended recipient can obtain the hidden data. The methods include invisible inks, microdots, character arrangement, digital signatures, covert channels, and spread spectrum communications. Steganography can be used to maintain the confidentiality of valuable information, to protect the data from possible sabotage, theft or unauthorized viewing (Westphal 2003).

Steganography has been widely used in historical times. Examples of these historical usage includes:

- Hidden messages in wax tablets: used in ancient Greece whereby people wrote messages on the wood, then covered it with wax so that it looked like an ordinary tablet.
- Hidden messages on messenger's body: used in ancient Greece. Herodotus tells the story of a message tattooed on a slave's shaved head, covered by hair growth, and exposed by reshaving. The message carried a warning to Greece about Persian invasion plans of the story is true.
- Hidden messages on paper written in secret inks under other messages or on the blank parts of other messages.



## 3.2 Digital Watermarking for Still Images

### 3.2.1 Introduction

Watermarking is becoming important in order to protect ownership rights. As the computers are more and more integrated via the network, the distribution of digital media is becoming faster, easier, and requiring less effort to make copies. Starting in the early 1990's, the World Wide Web became more and more popular, and offerings of multimedia resources delivered through digital networks became widespread (I. J. Cox & Bloom 2000). Therefore watermarking is a way to provide copyright protection for digital images, audio, video and multimedia products.

Companies that post their picture on the websites will like to ensure that no one can steal their image and post them on other Internet sites as users can easily download them from their sites. With digital watermarking, online content providers can embed watermarks in their files that flag the content as their property.

In conventional cryptography systems, once the encrypted data is decrypted, there is no way to track its reproduction or retransmission. Therefore conventional cryptographic techniques provide little protection against piracy.

Watermarks are digital signals or patterns that are embedded into digital signals (carriers). The watermark is usually not visible in the carrier signal by using the naked eye. Therefore the carrier signal alteration is not noticeable and strongly affected by such embedding. Since the watermarks are present in each unaltered copy of the original image, they can serve as a digital signature for the copies.

For the protection of digital products, a watermark can be used as a signature to signify ownership, and can only be detected by the legal owner.

They can also be used in order to provide image captioning. Illegally duplicated copies should not be able to remove the watermarks as this will cause serious degrade in the image. They must be robust to any product modification that does not degrade its quality. Resistance against any intentional attack is required (Pitas 1997).

### **3.2.2 Properties of Watermarks**

Digital watermarking hides data in a file by inserting a small amount of information throughout in such a way that the file can still be viewed. If the watermark is removed, the content of the media will be destroyed. When digital watermarking is applied, the purpose is to find information in the file that can be modified without having a significance impact on the actual content. Errors will be introduced into the content when a watermarked is applied. If the errors are low, the overall impact on the content will usually be minimal.

To classify a good watermarking technique, there are several criteria that a good watermark for an image must fulfill. These are as follows:

- Unobtrusive

A watermark is a perceptually unobtrusive signal embedded in an image, an audio or video clip, or any other multimedia asset. Its purpose is to be a label, which is attached to the content. The watermark signal should not affect or degrade the original image significantly. For the best result, the end user should not be able to distinguish any differences between the original and watermarked image by looking using their naked eye. The watermark should only be detected if the secret key is known (Kalker 1998).

- Robustness

Robustness refers to the ability to detect the watermark after common signal processing. For image, it includes spatial filtering, lossy compression, printing and scanning, and geometric distortions. Video watermarks may need to be robust to the same transformation as well as recording of video, changes in frame rate. Audio watermarks may need to be robust to process such as temporal filtering, recording on audiotape and variations in playback speed. The watermark must be difficult to remove and remain in the media content after the attack.

- Unambiguous

Retrieval of the watermark should clearly be able to identify the owner, and the accuracy of identification should degrade gracefully in the face of attack.

- Undeletable

The watermarks should be difficult to be removed by any hacker, without degrading the visual quality of the image.

- "Statistically Invisible"

To be statistically invisible means that the attacker is unable to detect the embedded message by comparing several different watermarks from the same author. This means that the watermark should not be obtained through statistical analysis on few different sets of watermarked data.

- Multiple Watermarking

The watermarking scheme should allow multiple data to be embedded into the same image for different authorized users. This data should be fully and unambiguously retrievable by the rightful owner with their corresponding use key.

### 3.2.3 Types of Watermarks

- Robust Watermarks

Robust watermarks are designed to resist against various removal methods. These watermarks are embedded in a way that any signal transformation of reasonable strength is not able to remove the watermark. When a robust watermark is designed, the watermark needs to survive processes that includes lossy compression, digital-to-analog-to-digital conversion, format conversion that are likely to occur during the embedding and detection (Kutter 2001).

- Fragile Watermarks

These watermarks can be detectable and destroyed with the slightest manipulations due to their low robustness. They can be used to check the reliability of objects and are comparable to the hidden messages in steganographic methods. However, a fragile watermark can be an advantage for authentication purposes. If a fragile watermark can be detected, it can be seen that there is not alteration done to the content since a watermark was embedded. A fragile watermarking scheme should be able to detect any changes in the signal and identify where it has taken place and what the signal was before modification (Kutter 2001).

- Public and Private Watermarks

These watermarks are differentiated according to the secret key used during the embedding and retrieving of the markings. The same key is used in the encoding and decoding process. The watermarks are referred to "public" if the key is known and "private" if the key is hidden. In private watermarking, only the authorized personnel are able to access the watermarks. Public watermarks are normally used in applications that do not have security-relevant requirements and can be detected by anyone (Bleumer 2004).

- Visible Watermarks

A visible watermark is a visible translucent image, which is overlaid over the primary image. These watermarks are not robust and can be used as logos or overlay images in the field of image or video watermarking (I. J. Cox & Bloom 2001).

- Invisible Watermarks

An invisible watermark is an overlaid image, which cannot be seen and can only be detected using the special algorithm. The watermark that is resistant to destruction under any image manipulation will be useful in verifying ownership of an image suspected of misappropriation. Digital detection of the invisible watermark would indicate the source of the image.

### 3.2.4 Applications of Watermarks

- Broadcast Monitoring

Broadcast monitoring used watermarks by putting a unique watermark in each video or sound clip prior to broadcast.

Automated monitoring stations can then receive broadcasts and look for these watermarks, identifying when and where each clip appears. This helps advertisers to ensure that the airtime purchased from the broadcasting firms is aired, musician and actors to ensure that they received accurate royalty payments for broadcast of their performance, and copyright owners to ensure that pirated stations do not illegally rebroadcast their property.

- Owner Identification

The copyright notice is still recommended for use today, although it is no longer necessary to guarantee copyrights. One disadvantage of text copyright notices is that they can often be removed from the protected materials when the Work is being cropped. Therefore digital watermarks can be used to provide complementary copyright marking functionality because it becomes an integral part of the content. The watermark can be made imperceptible and inseparable from the Work that contains them. Users that are supplied with the watermark detectors will be able to detect the embedded watermark that identifies the owner even after the Work is being modified in ways.

- Proof of Ownership

Watermarks are not just used to identify copyright ownership but also to proof ownership. Multimedia owners make use of watermarks to actually prove ownership. This is important when dispute arises whereby both parties claim to be the rightful owner. A person can easily steal the image, use an image processing program to replace the copyright notice with his own, and then claim to own the copyright. Therefore the use of embedding a watermark in the image can help to encompass the protection against

misappropriation of creations by other content providers without the permission of or compensation of the rightful owner.

- Authentication

As digital technology is increasingly used in both still and video cameras, the ability for undetectable tampering also increases. The use of authentication marks eliminates the problem of making sure that the signature stays with the Work but care must be taken to ensure that the watermark does not change the Work enough to make it invalid when compared with the signature. If one bit of a pixel of an image that is embedded with a cryptographic signature is modified, the tampering can be detected, as it no longer matches the signature. However, this signature is metadata that must be transmitted along with the photograph, in a header field of a particular image format. If the image is copied to another file format that does not contain this header field, the signature will be lost and the image can no longer be authenticated. Watermarking is used to embed the signature directly into the image. This eliminates the problem of ensuring that the signature stays with the image. This also make it possible for one to learn more about what tampering has occurred, since any changes made to the image will also be made to the watermark. There are several systems that are designed to indicate the estimated location of changes that had been made to the image.

- Transactional Watermarks (Fingerprinting)

Transactional watermarks, also called fingerprints, allow a content owner or content distributor to identify the source of an illegal copy. This capability allows a unique watermark to be embedded in each individual copy.

Electronic distribution of contents allows each copy distributed to be customized for each recipient. They can be embedded at the time of distribution to a specific customer. This requires a considerable computational overhead for the generation of watermarks and a medium that permits the efficient creation of distinct copies. Alternatively, a playback device that contains a subsystem tied to a specific individual can embed a fingerprint watermark immediately on playback. This approach reduces the computational burden of the content provider and does not require distinct copies. These fingerprints are potentially valuable as a deterrent to illegal use and as a technological aid to investigation.

- Copy Control

Transactional watermarking for monitoring, identification, and proof of ownership do not prevent illegal copying. They are powerful prevention and investigative tools. For copyright control, it is possible for recording and playback devices to react to an embedded signal so that a recording device might prevent recording of a signal if it detects a watermark that indicates recording is prohibited. However, for this system to work, all manufactured recorders must include watermark detection circuitry. Therefore watermarks help to provide protection against illicit use by end users (I. J. Cox & Bloom 2000).

### 3.2.5 Watermarking Techniques

- Spread Spectrum

Spread-spectrum communication includes a number of signaling techniques in which the transmitted bandwidth is significantly larger than required by the data rate. The transmitted bandwidth is determined by a function



independent of the message, and it be known to both the sender and receiver. In the spread-spectrum communication system, messages are encoded with a sequence of symbols. The symbols are transmitted in a temporal sequence, each one being represented by a signal referred as a "chip". These chips are pseudo-random sequences of 1s and 0s. In the frequency domain, they are spread over a wide range of frequencies. If some process distorts a fraction of the signal frequencies, a band-pass filter can be used to identify the chip.

- Quantization Index Modulation

Watermarking by quantization index modulation (QIM) proposed by Chen and Wornell (Chen & Wornell 2001) IEEE Transactions on Information Theory is one of the simplest non-linear methods to embed information based in a set of N-dimensional quantizers. This technique used the watermarked message as an index to select a particular quantizer from a set of possible quantizers (Chen & Wornell 2001). The message  $m$  that should be transmitted is the index for the quantizer used for quantizing the host-signal vector  $c_o$ . While retrieving the hidden information, one evaluates a distance metric to all quantizers. The index of the quantizer with the smallest distance contributes to the message  $m$ . To reduce distortion, the distortion constraint has to be fulfilled:  $E_k(c_o, m) = c_m \approx c_o$ . To increase the robustness, the reconstruction values of different quantizers must have a maximum distance.

- Patchwork Technique

This technique separates the data to be watermarked into two distinct subsets. One feature of the data is chosen, and modified in opposite directions in both subsets by labelling the sample values belonging to each subset.

The embedding and detection step is done using the separation of the samples. A test statistic can be defined that is compared against a threshold value. Therefore watermark can be easily detected if the data satisfies certain statistical properties (I. J. Cox & Bloom 2001).

- Least Significant Bit (LSB) coding

The least significant bit (LSB) method is based on the substitution of the LSB of the carrier signal with the bit pattern from the watermark. The bits are embedded in certain representation values, which the decoder will be able to retrieve the watermark if the values used for the embedding the individual bits are known. The substitution of the LSB is performed on the subset of all available carrier elements chosen by a secret key. During retrieving of the value of the bits, the decoder will need all the carrier elements that were used during the embedding process.

The random selection of the elements for embedding and the changing of the LSBs generate noise with low power and a constant power density. The perception of this noise depends on the perceptual threshold of the original carrier object, and also depends on its content.

## 3.3 Digital Watermarking for Audio

### 3.3.1 Introduction

In the mid 1990s, the initial audio watermarking research was inspired from image watermarking, as copyright protection is not a new issue. The concept was based on adding a watermark to the original audio signal and the watermarked signal

is perceived as identical to the original one by the listener.

At the same period, recording industry has been fighting against piracy. The digital revolution has brought this to a new level as music in digital format can be copied and distributed easily with no degradation. It can be easily distributed using electronic means such as the Internet with the aid of efficient compression algorithms (such as MP3) and peer-to-peer file sharing systems (such as Napster).

With watermarking, the watermark is embedded into the original audio signal without degrading the audio quality and should remain detectable and permanent. To offer copyright infringement, the compliant devices should check for the watermark before proceeding to operations.

### **3.3.2 Properties of Audio Watermarks**

The requirements that an audio watermarking system must satisfy are application dependent. All these requirements are to be respected to a certain extent according to the applications. Some applications might allow the watermark to introduce a small level of sound quality degradation while other would be extremely rigorous on that matter. Resistance to signal processing operations such as filtering, resampling and coding is usually necessary. For copyright protection, resistance to malicious attacks aimed at prevent watermark detection is also required. The general requirements are as follows:

- **Inaudibility**

The watermark embedded into the audio signal should not degrade the sound quality.

- Robustness

The watermark should resist any transformations applied to the audio signal as long as the sound quality is not unacceptably degraded.

- Capacity

The watermark bit rate must be high enough for the intended application, which can be conflicting with in audibility and robustness.

- Reliability

The data contained in the watermark should be extracted with acceptable error rates.

- Low Complexity

When the watermark is used for real-time applications, watermarking algorithm should not be excessively time-consuming.

### **3.3.3 Applications of Audio Watermarks**

- Proof of Ownership

When a piece of Work is composed, the artist will register the copy with the "trusted third party" (TTP) that acts as a repository of audio content before releasing the new Work. This is done to prevent an unauthorized artist to get a copy of the Work and releases it as his own that prevents the original artist to prove his ownership. A unique secret key, the owner's signature, is used to generate a watermark embedded into the audio signal, which must be accepted by a court of law as evidence of ownership.

This avoid the need of transferring the audio content itself to the TTP and new audio content is automatically protected if watermarked with the same key.

- Monitoring at the Customer End

With the implementation of monitoring applications at the consumer end, the aim is to avoid misuse of audio signals by the consumer. The watermark contains information that dictates the behaviour of compliant devices (such as CD players and recorders, MP3 players and computers) in accordance with the usage policy. This should prevent most home piracy, as the end user does not have the necessary skills to erase the watermark.

- Monitoring at Distributor End

Audio watermarking allowed all copyright recordings to be watermarked before released in order for server to check for the presence of the watermark. If no watermark is found, the recording is considered as copyright free. The advantage of this approach is the absence of a database as all the information necessary for system operation is carried by the watermarked signal itself. Therefore all copyright recordings are protected as individual recording has its unique watermark, which allow illegal copies to be traced.

- Identification of Broadcast Audio

Around the world, radio stations need to pay royalties for the music they aired. Rights holders need to monitor radio transmission in order to verify whether royalties are properly paid. Audio watermarking allowed all song and commercial to be identified. The watermark contains information that uniquely identifies the song and commercial. Therefore this also helps

advertisers to monitor radio and TV transmission to verify whether commercials are being broadcast as agreed.

- Tracking of Illicit Copies

Unauthorized use of copyright material has been common practice on the World-Wide Web. Audio watermarking can be used in audio file tracking systems. This approach consists of watermarking the recordings to be protected before distribution. Web crawlers can be used to automatically search the Web for the presence of watermark on each audio file it finds. If the watermark recording is found, the system notifies the owner who will contact the transgressor after confirming the infringement.

## **3.4 Attacks and Benchmarks of Digital Watermarking Systems**

### **3.4.1 Threats and Risks**

In each application, a certain level of security is always required. Competitors with the knowledge of the overall system will attempt to challenge the elements lacking in the mechanism due to the low security measures implemented. The first step is the identification of underlying risks and possible attacks that the system had to encounter. Therefore watermarking can be used to counteract some of these attacks. The watermarking operation consists of embedding, detection and removal of watermarks. The removal of watermarks is always an impermissible operation in security related applications for an attacker. Therefore the watermark has to be robust against processing manipulations, which can occur in the specific applications.

### **3.4.2 Classifications of Attacks**

In order to easily identify the attacks, a classification of the attacks into several groups helps both the developer of a watermarking algorithm and the user of the watermarking system in identifying the security requirements and judging the usability of the watermarking technology. This is important for the systems as the countermeasures for some attacks are still not reliable.

When a watermark is embedded, a detection of the watermark always implies. Three major categories of effects making watermarking useless during the detection is identified:

- Watermarks cannot be detected. The watermark might be removed or misaligned.
- False watermarks are detected. This can be accomplished by attacks that perform some kind of embedding of false watermarks.
- Unauthorized detection of watermarks. Algorithms that are not carefully designed can produce false alarms.

Different types of attacks are possible depending on the knowledge of attackers and the tools acquired. The overview of all possible attacks is shown in Table 3.1, group according to the results of the attacks and the assumptions about the attacker.

Effect Operation	No Detection Remove/Desynchronize	False Detection Embed	Unauthorized Detection Detect
No Knowledge	Signal processing/ misalignment	Copy attack	-
Algorithm published	Specific designed attacks	Deadlock attack	-
Marked works	Collusion attacks	Copy attack	-
Detector	Oracle attacks	Copy attack	False alarms
Encoder and detector	Custom-tailored oracle attack	Overmarking	False alarms

Table 3.1: Table of Attacks

Each row corresponds to a different assumption about the attacker and represents a variation of one category. The three columns of the table represent the major classes of attacks. The class of attacks that produces the "no detection" result is further subdivided into two classes, removal attacks and desynchronization attacks, according to the way the intended effect is achieved.

From the table, attacks requiring no prior knowledge are the most general form and usually based on common signal processing operations. Collusion attacks is having access to watermark copies of the same work with different watermarks or different works with the same watermark. When the detector is available, sensitivity analysis can be applied. If both the embedder and detector are available, attacks like custom-tailored oracle attack can be applied.

#### 1. Removal Attacks and Manipulations

The removal of watermarks represents the most obvious form of attacking a watermark. The restoring of the original will be very complex. If attackers have no prior knowledge of an algorithm, the watermarks will be subjected to distortions. The removal of watermarks can also happen unintentionally due to operations during the preprocessing of the data in certain applications.



- Signal processing operations

Signal processing manipulations can be used in order to remove watermarks. Users with no special knowledge of signal processing can also apply these operations by using common consumer grade software products to perform filtering and compression operations automatically. This manipulation will be more critical if a detailed procedure for removing watermarks is widely distributed.

## 2. Desynchronization Attacks

The aim of desynchronization is to make the embedded watermark undetectable. The process of detecting the watermark by desynchronization attacks is different. Instead of erasing the watermark, misalign the embedded watermark and detector in a way that it is infeasible to perform synchronization prior to detection.

- Global and local transformations

Most of the watermarking algorithms require near perfect alignment during detection. Therefore, applying global and local transformations aims at the destruction of the synchronization between the watermark and the detector. Global distortions of watermarked include shifting, rotation, and scaling for images and video and time scaling for audio creations.

- Scrambling attacks

Scrambling attacks is another kind of desynchronization by scrambling samples of the watermarked creation prior to the presentation to a watermark detector. If the watermark are not directly modified but only their presentation, the attacks are performed on a system level that cannot be addressed within the watermarking system itself.

### 3. Embedding Attacks

Embedding attacks simulate an embedded watermark even if it is not embedded. The effect of this attack is the false detection of watermarks.

- Copy attack

The aim of copy attack is to copy a watermark from one carrier signal to another. This attack basically performed an estimation of the watermark calculated from the marked carrier signal. The estimated watermark signal is then copied from the marked signal to the target carrier data to obtain the watermarked version. The estimated watermark can be obtained in different ways depending on the assumptions made about the attackers. If there is no prior knowledge of the algorithm but has access to the same object carrying different watermarks, a collusion attack can be performed to approximate the original object. The estimated watermark is obtained by subtracting the estimated original from the corresponding watermarked version.

- Overmarking

Overmarking is an operation where a second watermark is embedded in an already marked carrier signal. The secret key can detect both watermarks independently. This operation can be performed if the attacker has access to the embedder and detector of the watermarking system.

### 3.4.3 Benchmarking

Benchmarking is normally used to evaluate when one of the established or emerging techniques is superior to the available alternative methods. Watermarks

algorithms are mainly judged by their ability to preserve the quality of the original carrier signal and the robustness of the embedded watermarks. A watermarking benchmark is used for different reasons listed as follows:

- All watermarking algorithms have individual strengths and weaknesses that must be taken into consideration by a potential user in evaluating a given system for an application.
- Watermarking systems for developers have an interest in judging the relative and absolute merit of new techniques or variations on existing ones. They might also be interested in detecting weaknesses for future algorithm improvements.
- Watermarking system vendors are potentially interested in an objective and independent comparison of available commercial system.

These scenarios represent different approaches to the use of a benchmark system. Thus, different conclusions must be drawn to enable the development of a benchmark system that is able to cover all the different aspects of possible users.

- A benchmark system must have well-defined, realistic scenarios. These scenarios are the basis for the evaluation of watermarking algorithms. A variety of different scenarios are provided for a benchmark system, which must be highly correlated with real world applications in terms of attacks as well as test data used in the benchmark.
- A benchmark system must be independent of developers and vendors. A third party with no conflicting interests should have developed the benchmark system. During the development stage, all ideas and aspects of watermarking developers and users should have been considered. A third party

should be able to perform the benchmarks in the suite under controlled circumstances and supervise the system under test to prevent alterations and manipulations.

- The results and reports must be clear and significant. A ranking score might be helpful but such ranking will depend on the time of execution of the benchmark because of its dynamic adaptation unless an absolute metric can be established. Therefore, time stamp of the test scenarios are necessary. This is important to achieve reproduceability of the test results, since the benchmark suite is likely to evolve even in case absolute metrics are used, resulting in incomparable results unless versions are taken into account.

## Chapter 4

# Experimental Investigation of Watermarking Robustness

### 4.1 Algorithm

A variety of watermarking techniques already exist to embed information into digital media content. The techniques range from LSB to spread-spectrum method. With the knowledge gained from the different watermarking techniques, the least significant bit (LSB) watermark technique is particularly studied and implemented in this project.

LSB watermarking is a technique to embed the watermark into the least significant bit of the cover object. It is based on the substitution of the LSB bit of the carrier signal with the bit pattern from the watermark. The bits are embedded in certain representation values such as pixels. The decoder in turn is able to retrieve the watermark if it has the knowledge of the of the representation values used for embedding the individual bits.

The watermark encoder uses a subset  $c_{oj}[1], \dots, c_{oj}[(c_{oj})]^5$  of all available carrier elements  $c_o$  chosen by the secret key. The substitution operation  $c_{oj}[i]-m[i]$  on the LSBs is performed on this subset. The reading process retrieves the values of these bits. Therefore, the decoder needs all the carrier elements that were used during the embedding process.

## 4.2 Test Images

The bitmap images and watermark image used in the project are displayed in monochrome with 256x256 pixels and 12x9 pixels respectively. These bitmap images are some of the commonly used test images. A number of test images are chosen for the LSB watermarking technique so that the behaviour of the watermark being embedded to and decoded from the images can be examined. Monochrome images are used for the ease of extracting the data. The bitmap images and watermark image are as shown:



Figure 4.1: Test image : bird.bmp (256x256 pixels)



Figure 4.2: Test image : camera.bmp (256x256 pixels)



Figure 4.3: Test image : lena.bmp (256x256 pixels)

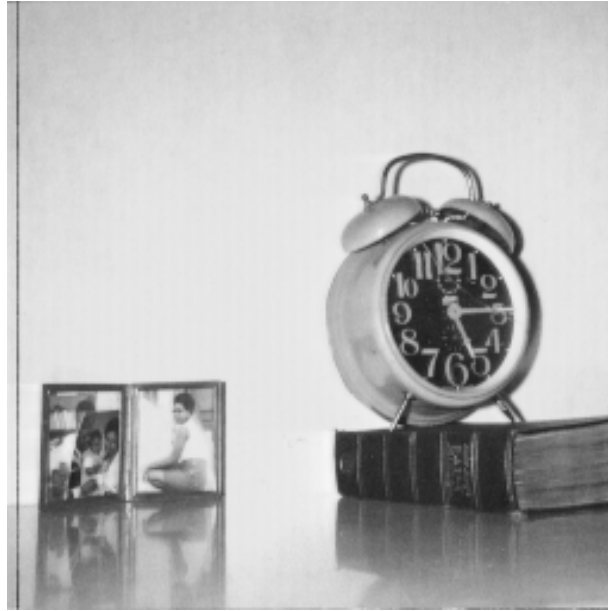


Figure 4.4: Test image : clock.bmp (256x256 pixels)





Figure 4.5: Test image : bridge.bmp (256x256 pixels)

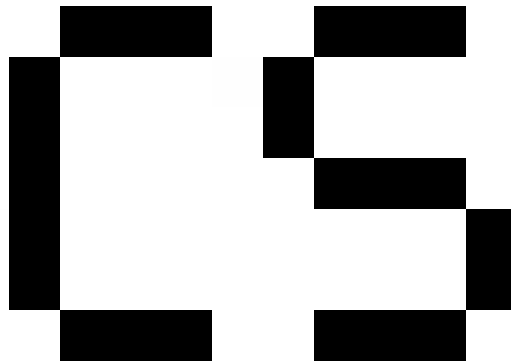


Figure 4.6: Test image : copyright.bmp (12x9 pixels)

---

## 4.3 Software

### 4.3.1 MATLAB

The programming tool used in the project is MATLAB 6.1. MATLAB was selected because of the familiarity in the commands used and is previously installed in the computer as required for the used for other subjects during the whole degree. A good understanding of the build-in routines and codes is needed in order to meet the programming requirements of the project. The built in functions for handling bitmap files would simplify the development of the code. The *imread* command is specifically for dealing with the bitmap images. The availability of this command allows easier and faster development of the code.

All the programming scripts for embedding and decoding of the watermark are done using MATLAB.

GUI menus are also created using MATLAB. This allows the user to test the LSB watermarking technique in a friendlier environment instead of having to run all the MATLAB code individually.

### 4.3.2 IrfanView

IrfanView is a very fast, small, compact and innovative freeware graphic viewer for Windows 9x/ME/NT/2000/XP/2003. In this project, the program is used to compress the bitmap watermarked image into the GIF format. The GIF watermarked image is then decompressed by the program to the bitmap format for watermark retrieval.

## 4.4 Program

### 4.4.1 MATLAB code

The MATLAB source code of the programs are shown in Appendix C. The description of the function of different .m source code will be explained below

- Main.m



Figure 4.7: Main.m : LSB watermarking main menu)

The main.m source code is the core program to run for the project. This program generates a menu with the start and exit button. When the 'Start' button is chosen, the user will be led to the main1.m menu. If the 'Exit' button is chosen, the LSB watermarking technique will end.

- Main1.m



Figure 4.8: Select image menu

The main1.m generates a menu for the user to select the bitmap image as a test image. This test image will be watermarked in the LSB watermarking technique. If different image buttons are chosen, the program will proceed to main2.m menu. When the 'Return to Main Menu' button is chosen, the program will return back to the main.m menu.

- Main2.m



Figure 4.9: Select watermarking scheme menu

A menu that allows the selection of the number of watermarks to be embedded into the test image is generated in main2.m. When the 'Single Watermark' button is selected, a single watermark will be embedded into the test image. If the 'Multiple Watermark' button is chosen, 8 watermarks will be embedded into the test image. The 'Return to Image Selection' button will bring the user back to the main1.m menu.

- Main3.m

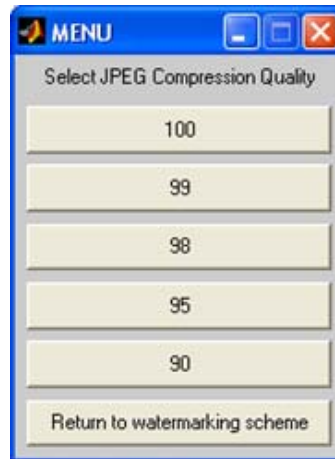


Figure 4.10: Select JPEG compression quality menu

The main3.m menu lets the user select the JPEG compression quality after the user has embedded the watermark using the preferred scheme selected in main2.m. The main3.m will then compress and save the watermarked image with the user's choice of quality factor. The command used to save the image is *imwrite*. The 'Return to Watermarking Scheme' button will bring the user back to main2.m to reselect the watermarking scheme.

- Main4.m

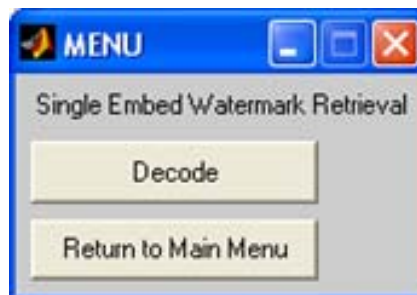


Figure 4.11: Single embed watermark retrieval

The main4.m menu is used to decode the watermark from the compressed watermarked image. This menu will only be executed if the single watermark embedding scheme is chosen in main2.m. The program will return to the main.m menu if the other button is selected.

- Main5.m



Figure 4.12: Multiple embed watermark retrieval

The watermark can be retrieved from the compressed watermarked image with or without the used of a comparator in the main5.m menu. The program is executed when the multiple embedding of watermark is chosen in main2.m. The 'Return to Main Menu' button will return to main.m menu.

- Single\_embed.m

The program will firstly read in the bitmap test image and watermark image using the *imread* command. The size of both the images are obtained and then converted from an  $M \times N$  matrix into a single row array. A set of random numbers is generated according to the size of the watermark image. These numbers will select the pixels of the bitmap image to be embedded with the watermark pixels. The LSB of the selected pixels will then perform a *bitxor* operation with the watermark image pixels.

After that, the watermarked image pixels will be converted from a single row array back to the  $M \times N$  matrix and compressed with the quality factor chosen by the user. The bitmap image, watermark image and watermarked image are plot using the *image* command.

- `Multitple_embed.m`

The *imread* command read in the bitmap test image and watermark image into the program. The *size* command is then used to determine the dimension of these images. The bitmap image is first converted from an  $M \times N$  matrix into 8 smaller  $A \times B$  matrixes, and then each smaller  $A \times B$  matrix is converted to a single row array. The watermark image is also converted to a single row array. The size of the watermark image is used to generate 8 sets of random numbers. Each set of random numbers is used to choose the pixels of the each smaller bitmap image to be embedded by the watermark image. The *bitxor* operation is then performed to embed the watermark pixels into the LSB of the selected image pixels. Therefore 8 watermarks will be embedded into the bitmap image. After the embedding process, all the smaller  $M \times N$  matrixes are combined back into an  $M \times N$  matrix. The watermarked image is compressed with the quality factor selected. The *image* command is used to plot the bitmap image, watermarked image and watermark image.

- `Single_decode.m`

The compressed watermarked image is read into the program and its dimension is determined. The  $M \times N$  compressed watermarked image is then converted to a single row array. The *bitxor* operation is performed on the compressed watermarked image and the bitmap image in order to retrieve the back the watermark pixels.



The decoded watermark pixels are converted into a matrix and plotted using the *image* command.

- Without\_comparator.m

The size of the compressed watermarked image is measured after read into the program. The MxN compressed watermarked image is then converted into 8 smaller AxB images. The watermark pixels are then retrieved using the bitxor operation on each AxB image with the AxB bitmap image. Therefore 8 watermark pixels will be decoded. The final value of the watermark pixels are determined by taking the average of the sum of all the watermark pixels retrieved. The watermarked pixels are converted back into a matrix and plotted.

- With\_comparator.m

The compressed watermarked image is converted into 8 smaller AxB images after being read into the program and after the size is determined. Each AxB compressed watermarked image will perform a bitxor operation with the AxB bitmap image. The watermark pixels will then be decoded and there will be 8 sets of data retrieved. The final watermark pixels will be determined with the used of a comparator designed for this project. The comparator will determine the most frequent value that appears on each pixel positions and take it as the final pixel value. When all the watermark pixels are decoded, the watermark is plotted after converted back into an MxN matrix.

---

#### 4.4.2 Flowchart of Source Code

The flowchart for different MATLAB scripts in this project are as shown below. It consists of the overall program flowchart, the embedding of the watermark using the single watermark and multiple watermarks flowchart, and decoding of watermark using the single watermark decoder, and with or without comparator multiple watermark decoder.

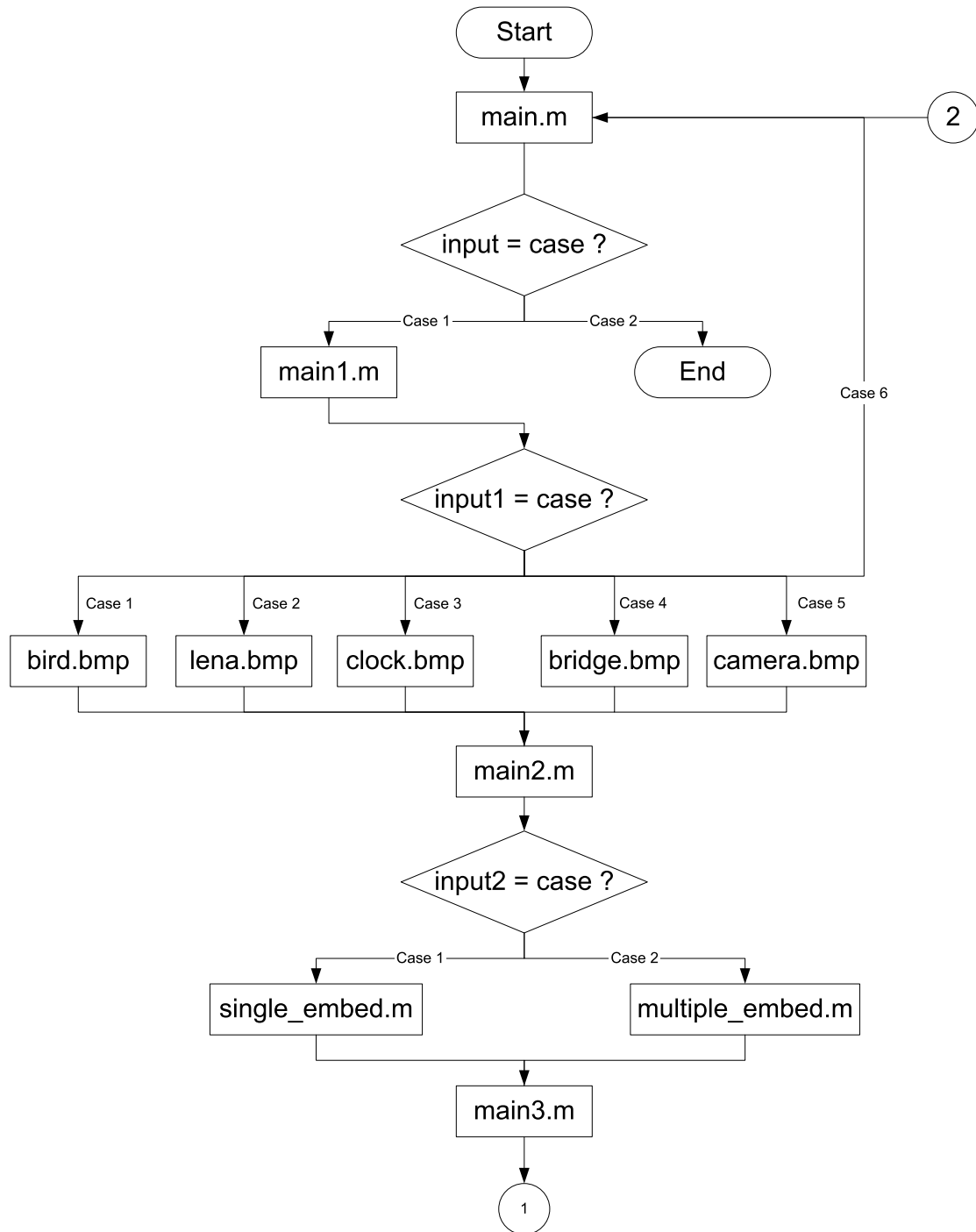


Figure 4.13: Flowchart of overall program

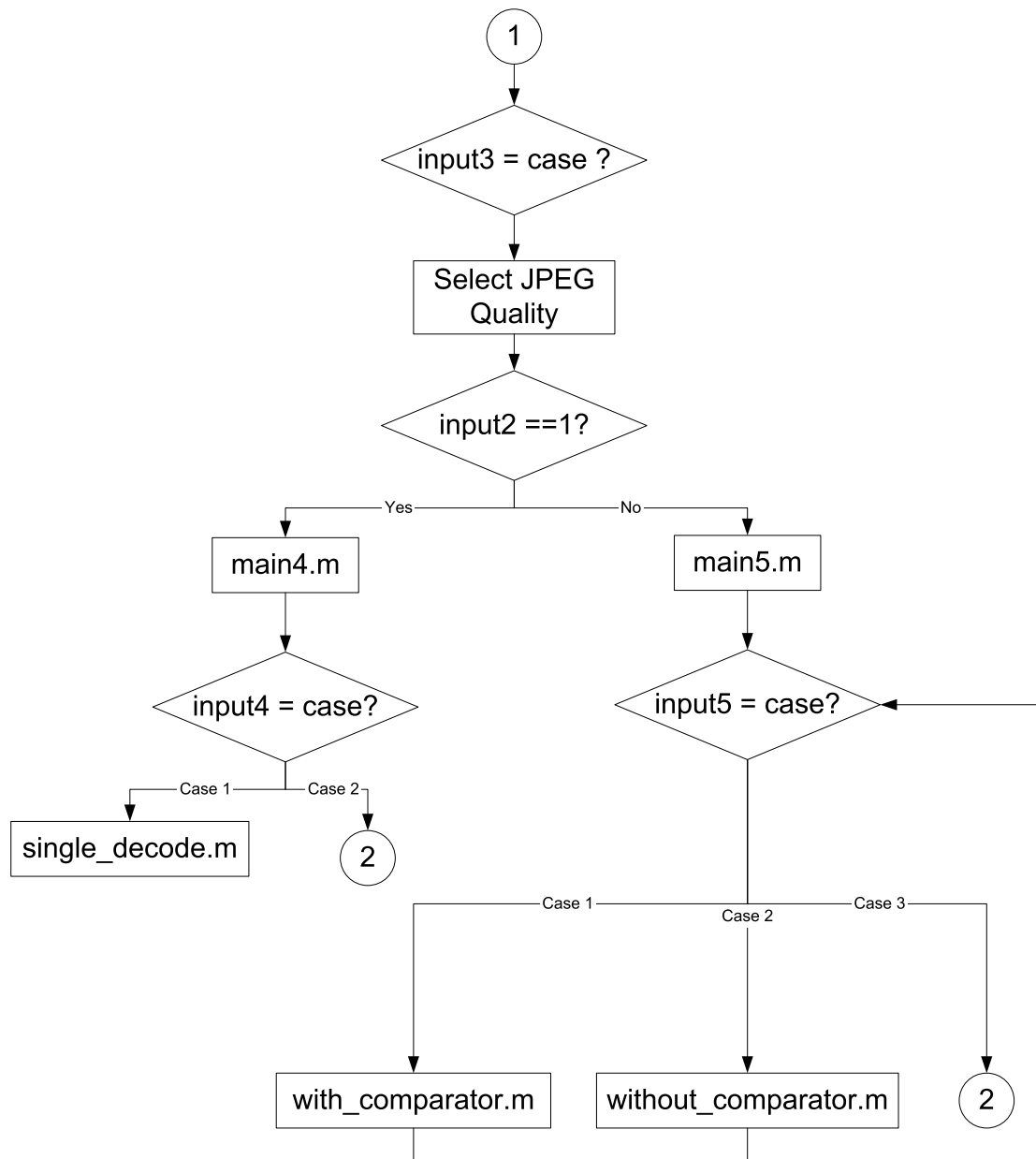


Figure 4.14: Flowchart of overall program

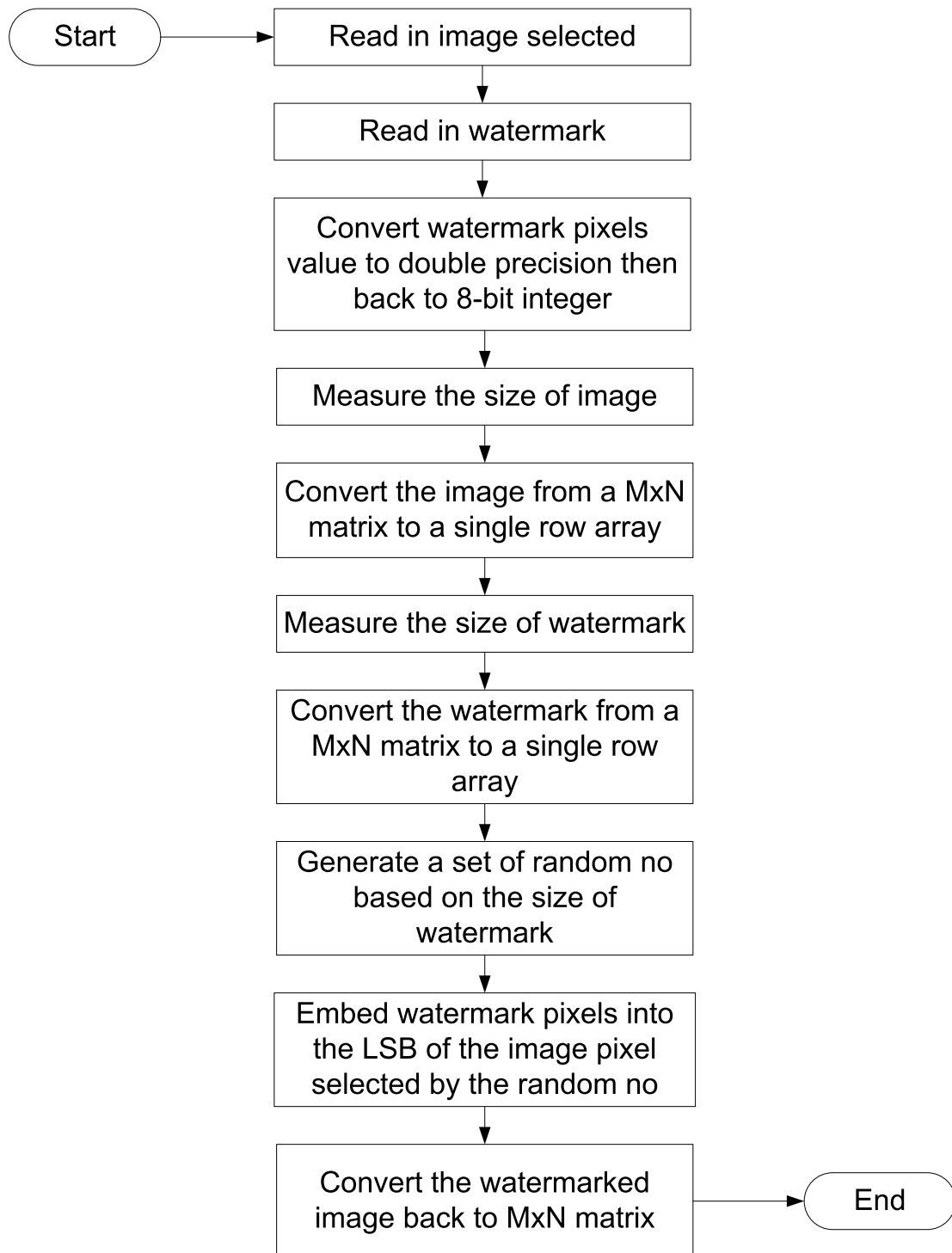


Figure 4.15: Flowchart of single watermark embedding

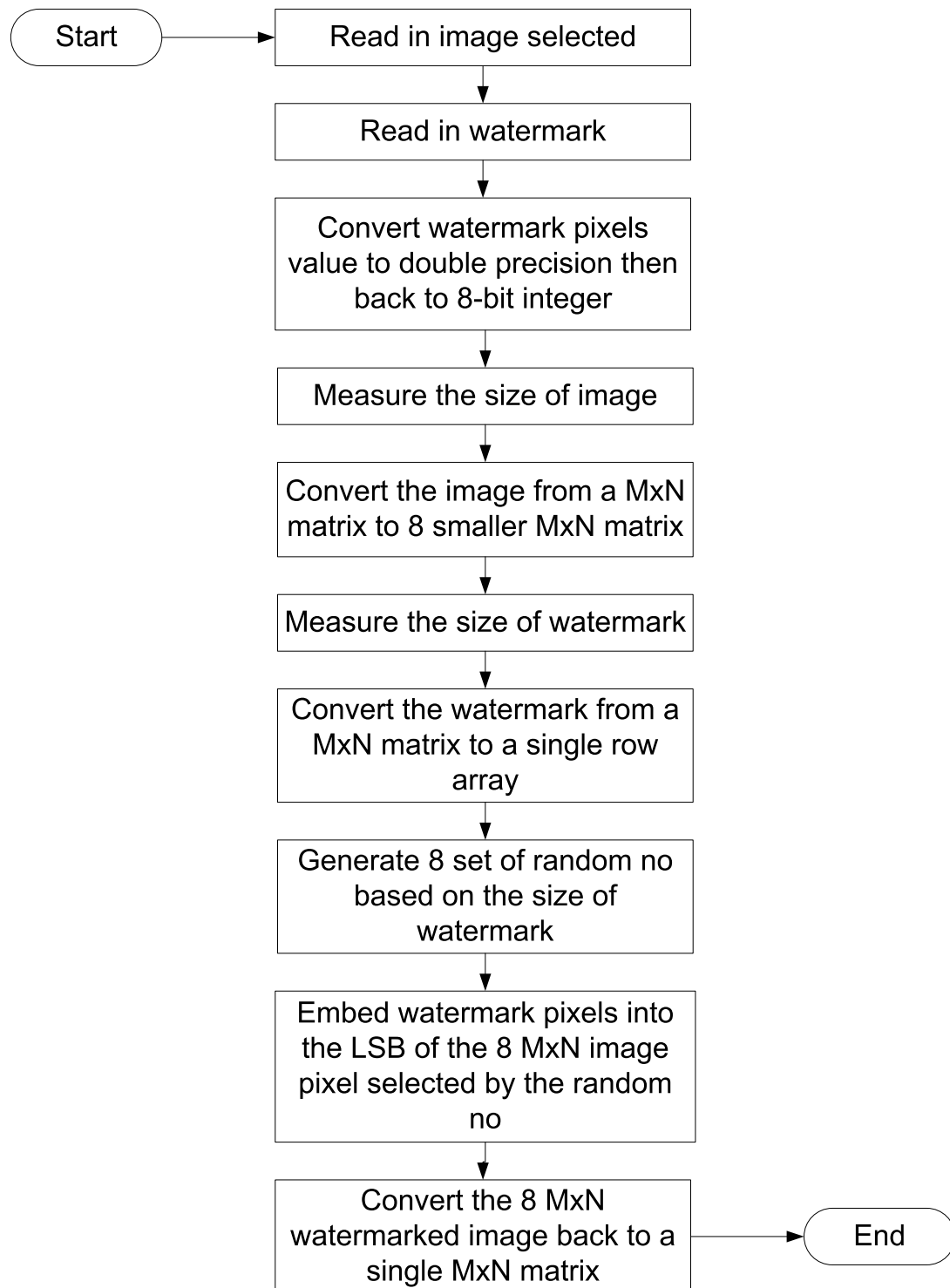


Figure 4.16: Flowchart of single watermark embedding

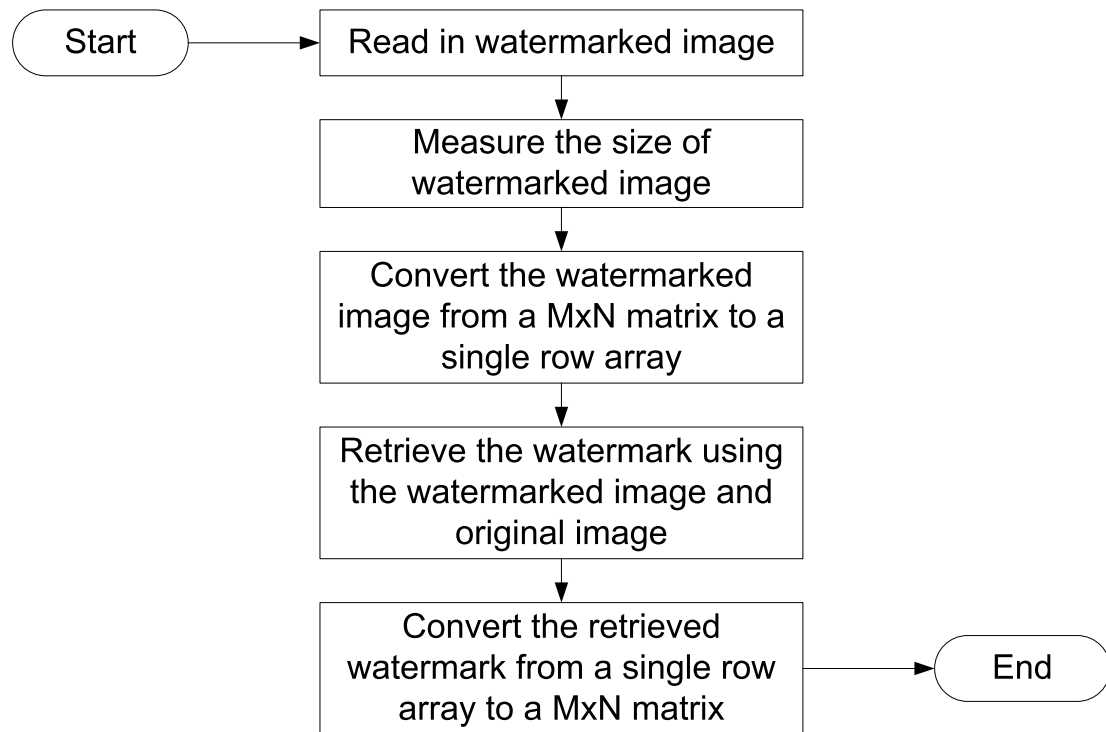


Figure 4.17: Flowchart of single watermark decoding

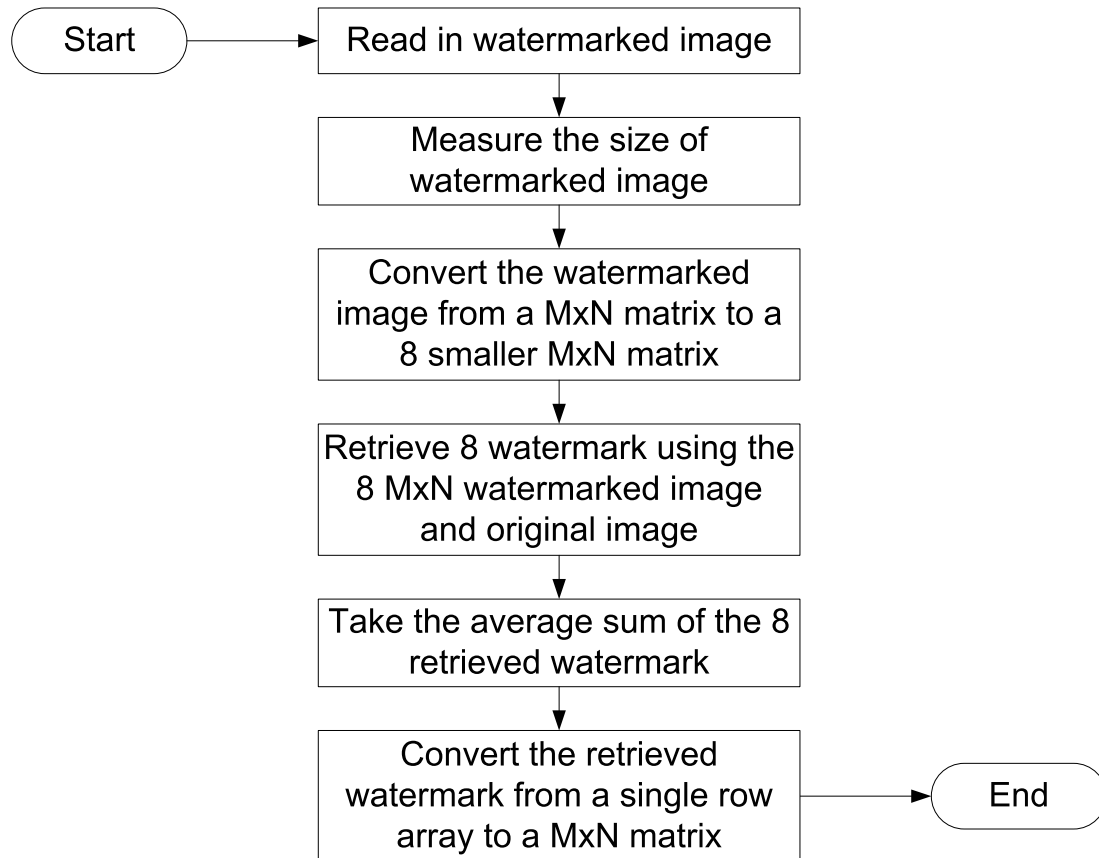


Figure 4.18: Flowchart of multiple watermark decoding without comparator



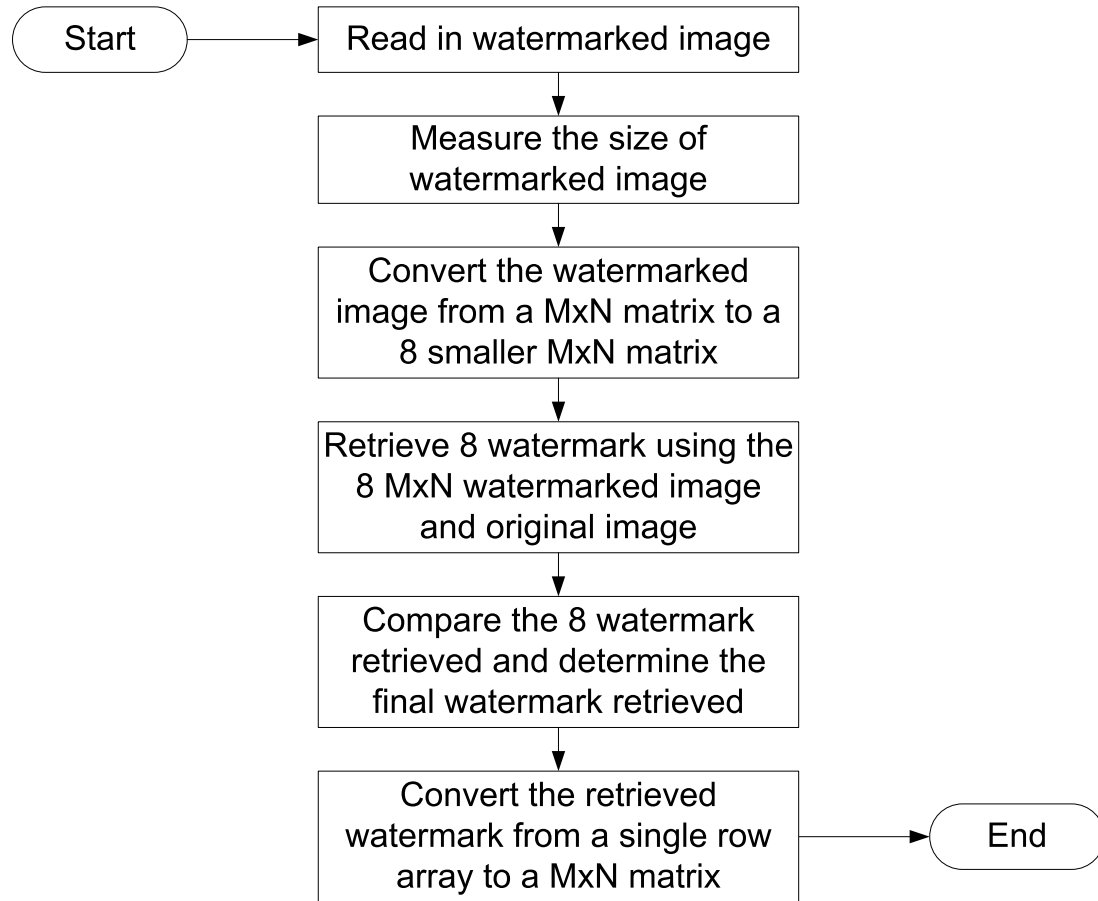


Figure 4.19: Flowchart of multiple watermark decoding without comparator

#### 4.4.3 Graphical User Interface (GUI)

The Graphical User Interface (GUI) is programmed using the *menu* command in MATLAB. This allow the user to use and test the program is a friendlier environment as there is no necessity to ponder which program is for what function. All the MATLAB scripts with different functionalities are link together with the use of different GUI. The GUI chart is as shown in Figure 4.20:

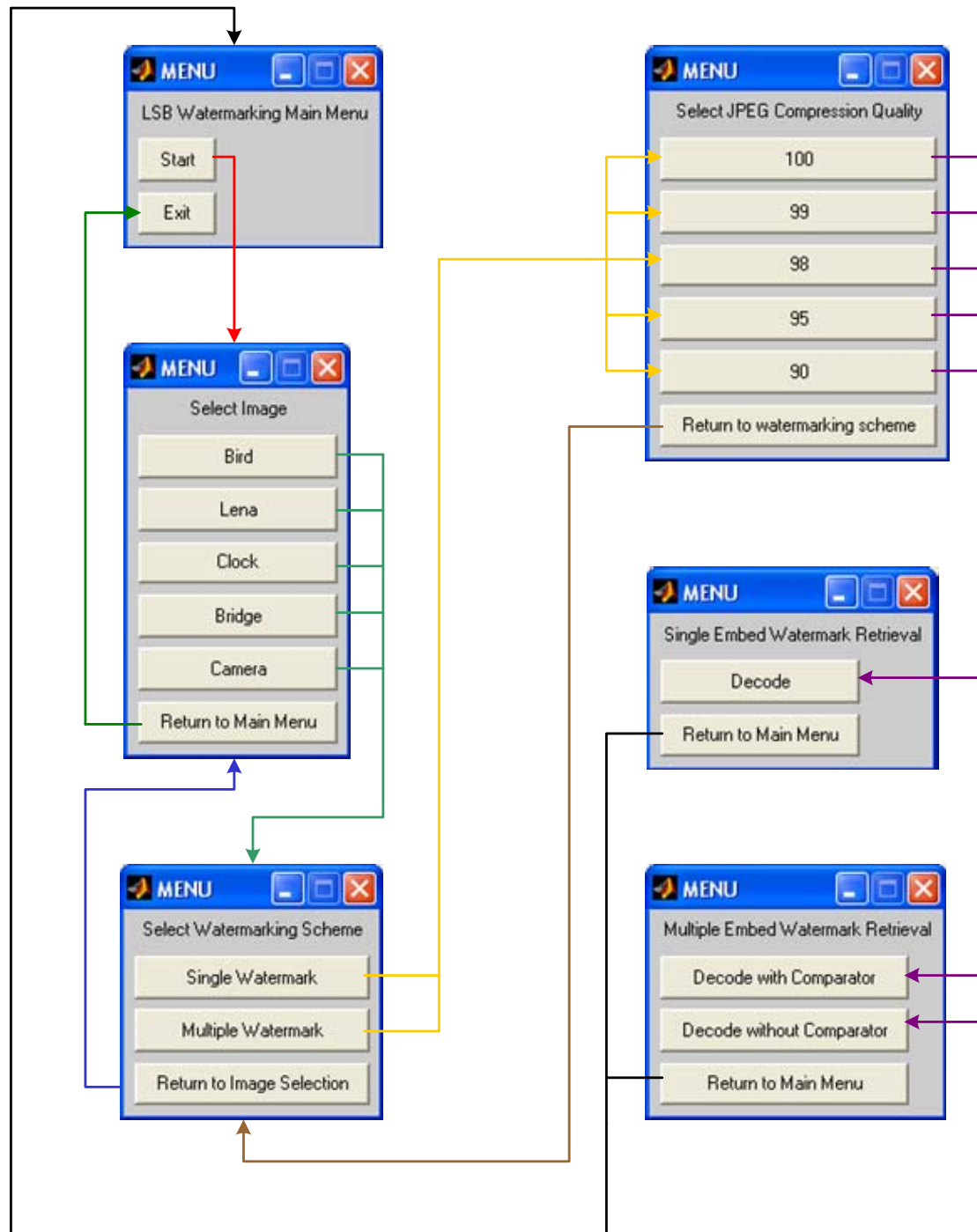


Figure 4.20: Graphical User Interface (GUI)

---

## 4.5 Results

The results of the GIF and JPEG compression techniques will be discussed. The discussion will be based on the different test bitmap images that are used in the LSB watermarking technique. The graphical display of retrieved watermarked using different compression quality will be presented. The detail test results are shown in Appendix B.

### 4.5.1 Joint Photographic Experts Group (JPEG)

JPEG is a compression technique for colour images and photographs that balances compression against loss of detail in the image. It is a popular file compression format which allows the storage of high quality images in relatively small files. JPEG is also called lossy compression as more information is lost when the compression is greater. JPEG deletes information from an image that it considers unnecessary that can range from small amounts of lossless compression to large amounts of lossy compression.

- Bird

From Figure 4.21, the decoded watermark images result from using the single watermark embedding are not identical to the original image. At 100%, only a fairly similar watermark image can be retrieved. As the compression quality further decreases, the watermark images retrieved are unrecognizable compared to the original watermark image. This can be seen that the single watermark embedding and decoding are not susceptible to JPEG compression.

When multiple watermarks are embedded into the different positions of the bitmap test image, the watermark image retrieved with or without the comparator can be seen in Figure 4.21. The watermark image retrieved without using the comparator with 100% quality factor is slightly better than the single watermark decoding. But the watermark retrieved when the compression decreases is losing all the black pixels. At 99% and 100% compression quality factor, the comparator used in the multiple watermark is able to decode identical watermark image. The watermark images retrieved are also unrecognizable when the compression quality further decreases.

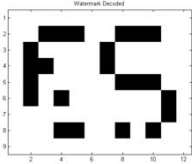
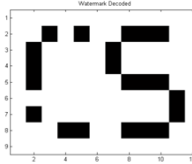

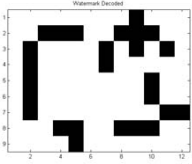
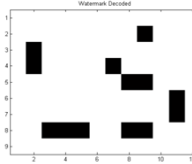

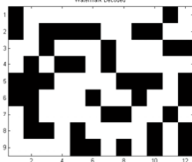
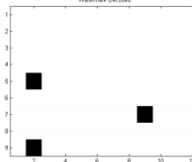
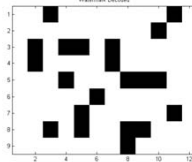
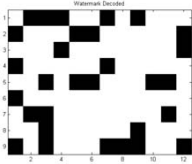
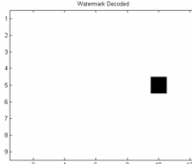
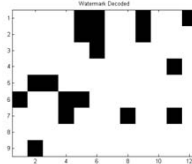
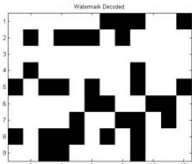
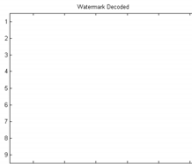
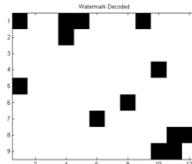
Bird	Quality Factor	Single Watermark	Multiple Watermark	
			Without Comparator	With Comparator
	100%			
	99%			
	98%			
	95%			
	90%			

Figure 4.21: Test results for Bird.bmp

- Lena

The watermark images retrieved from the single watermark decoding are shown in Figure 4.22 are not similar to the original watermark for the entire range of compression quality factor tested.

The multiple watermarks implementation without a comparator is able to retrieve a watermark image almost similar to the original image at 100% compression quality. As the compression quality factor decreases, the watermark images retrieved are losing almost all the black pixels information. The comparator is able to retrieve an identical watermark image as the original at 100% and an almost similar watermark image is also retrieved at 99%. At other compression quality, the watermark images retrieved are different from the original watermark.

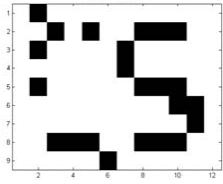
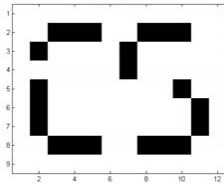
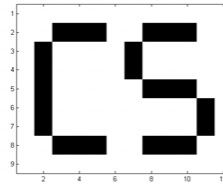
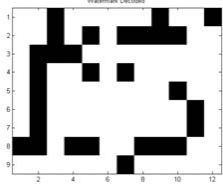
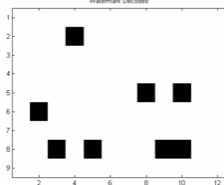
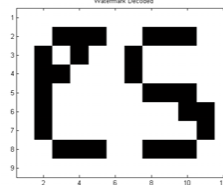
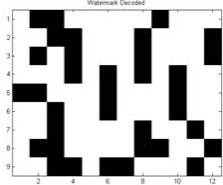
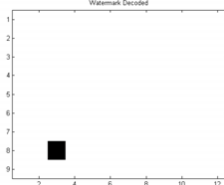
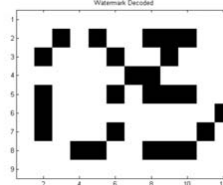

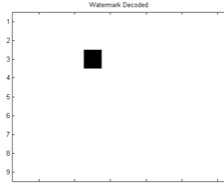

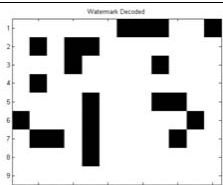

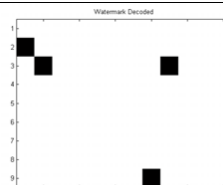
Lena	Quality Factor	Single Watermark	Multiple Watermark	
			Without Comparator	With Comparator
Lena	100%			
	99%			
	98%			
	95%			
	90%			

Figure 4.22: Test results of Lena.bmp

- Clock

From Figure 4.23, a comparable watermark image is retrieved at 100% compression quality using the single watermark. The retrieved watermark images for the other compression quality factor are dissimilar from the original watermark completely.

When multiple watermarks without a comparator is applied, the watermark images retrieved are totally different from the original watermark. As the compression quality factor decreases, the values of the black pixels are all missing in the retrieved watermark images. An identical watermark is retrieved at 100% compression quality after a comparator is applied to the multiple watermarks scheme. The watermark images retrieved then slowly become more dissimilar to the original watermark image as the compression quality factor decreases further.



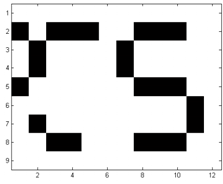
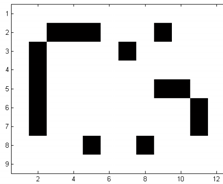
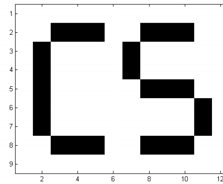
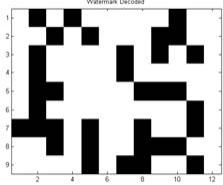
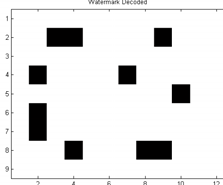
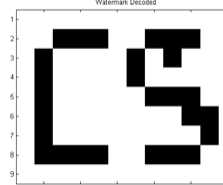
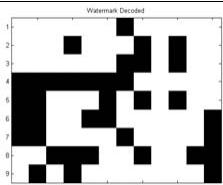
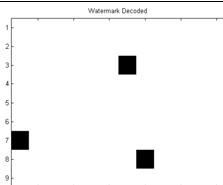
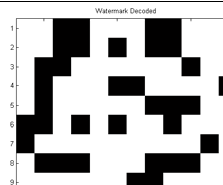

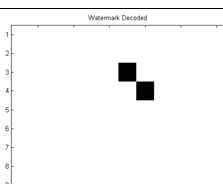
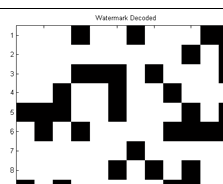
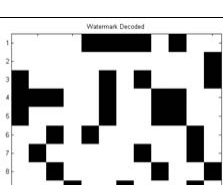
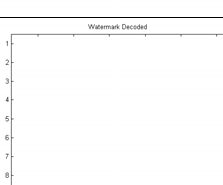
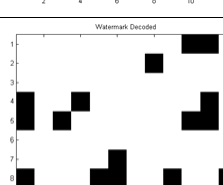
Clock	Quality Factor	Single Watermark	Multiple Watermark	
			Without Comparator	With Comparator
	100%			
	99%			
	98%			
	95%			
	90%			

Figure 4.23: Test results of Clock.bmp

- Bridge

In Figure 4.24, the single watermark is not able to retrieve any watermark image identical to the original watermark for the range of compression quality factor except at 100%. This is the same for the multiple watermarks without comparator. When the compression quality is at 98% and decreasing, most of the black pixels value is lost.

With the comparator, an identical and almost similar watermark images are retrieved at 100% and 99% compression quality factor respectively. The watermark images retrieved after 98% are all completely different.

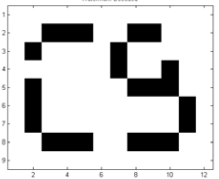
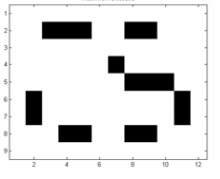
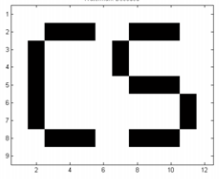
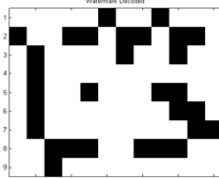
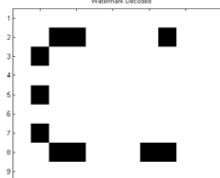
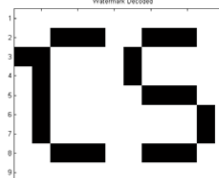
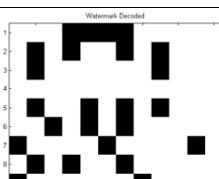
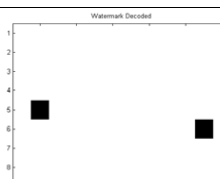
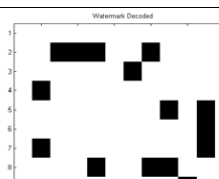
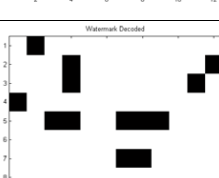
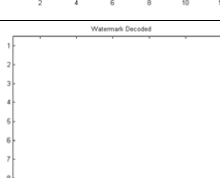
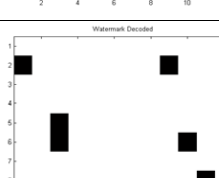
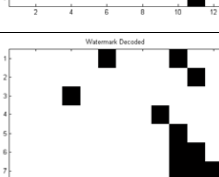
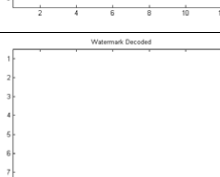
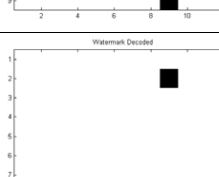
Bridge		Single Watermark	Multiple Watermark	
Quality Factor			Without Comparator	With Comparator
100%				
99%				
98%				
95%				
90%				

Figure 4.24: Test results of Bridge.bmp

---

- Camera

All the retrieved watermark images using single watermark are unlike the original watermark as shown in Figure 4.25. When multiple watermarks without comparator is used, the retrieved watermark image at 100% is almost similar. The watermark images retrieved at other compression qualities are different and almost losing all the black pixels values. At compression quality of 100%, an identical watermark is retrieved. After which, the watermark images retrieved are getting even more different from the original watermark image.

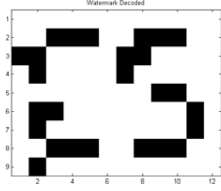

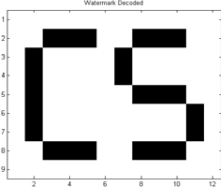
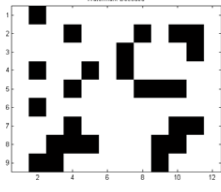
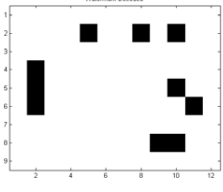
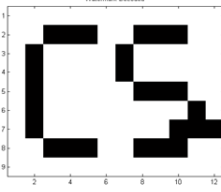
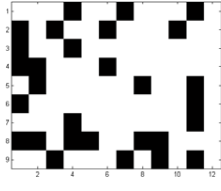
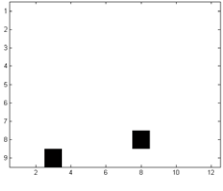
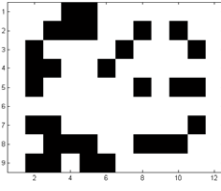
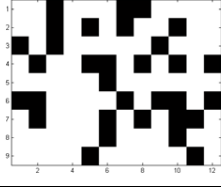
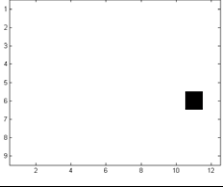
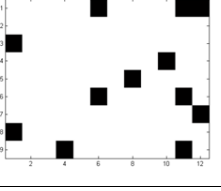
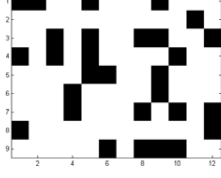
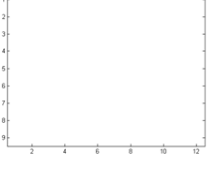
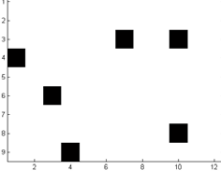
Camera	Single Watermark	Multiple Watermark	
Quality Factor		Without Comparator	With Comparator
100%			
99%			
98%			
95%			
90%			

Figure 4.25: Test results of Camera.bmp

In Table 4.1, the file sizes of the different bitmap test images after subjected to different compression quality factor are shown. The compression ratio is also calculated to show the amount of information is being compressed. It can be seen that the compression ratio of different bitmap test images are different because of the different information contained in each image.

<b>Original bitmap test image size = 65kbps</b>						
<b>Quality Factor</b>	<b>Image</b>	<b>Bird</b>	<b>Lena</b>	<b>Clock</b>	<b>Bridge</b>	<b>Camera</b>
<b>100%</b>	<b>File Size (kbps)</b>	<b>34.2</b>	<b>45.4</b>	<b>36.6</b>	<b>59.8</b>	<b>44.6</b>
	<b>Compression Ratio (%)</b>	<b>52.6</b>	<b>69.85</b>	<b>56.31</b>	<b>92</b>	<b>68.62</b>
<b>99%</b>	<b>File Size (kbps)</b>	<b>31.8</b>	<b>42.6</b>	<b>33.8</b>	<b>56.1</b>	<b>41.6</b>
	<b>Compression Ratio (%)</b>	<b>48.92</b>	<b>65.54</b>	<b>52</b>	<b>86.31</b>	<b>64</b>
<b>98%</b>	<b>File Size (kbps)</b>	<b>26.4</b>	<b>37</b>	<b>28.4</b>	<b>49.1</b>	<b>35.6</b>
	<b>Compression Ratio (%)</b>	<b>40.62</b>	<b>56.92</b>	<b>43.69</b>	<b>75.54</b>	<b>54.77</b>
<b>95%</b>	<b>File Size (kbps)</b>	<b>17.7</b>	<b>26.8</b>	<b>20</b>	<b>38</b>	<b>25.7</b>
	<b>Compression Ratio (%)</b>	<b>27.23</b>	<b>41.23</b>	<b>30.77</b>	<b>58.46</b>	<b>39.54</b>
<b>90%</b>	<b>File Size (kbps)</b>	<b>11.9</b>	<b>18.8</b>	<b>13.8</b>	<b>28.8</b>	<b>18</b>
	<b>Compression Ratio (%)</b>	<b>18.31</b>	<b>28.92</b>	<b>21.23</b>	<b>44.31</b>	<b>27.69</b>

Table 4.1: JPEG compression quality table for single and multiple watermarks

From test results in Figure 4.21 to 4.25 and Table 4.1, the different compression quality factor will result in different file sizes and compression ratios on the bitmap test images. It can be seen that the single watermark implementation is not robust to compression techniques as all the watermark images retrieved are not identical to the original watermark image.

These shows that the watermark that is embedded into the bitmap test image is being altered during the compression process.

Multiple watermarks are also embedded into the bitmap test images. When the watermark images are decoded without a comparator, the retrieved watermark images are almost similar to the original watermark at 99% and 100% compression quality factor. As the compression quality factor decrease further, it can be observed that the retrieved watermark images are losing almost all the black pixels value. Therefore, by taking the average sum of all the watermark retrieved is not a good technique and will even result in more pixels information lost when the compression quality factors gets lower.

A comparator is implemented in the multiple watermark scheme. This comparator will determine the final value of the pixels by comparing all the retrieved watermark images pixels. The final pixels value will be based on taking the most frequent pixels value that appear at that particular position. From the test results, it can be seen that the watermark image can be retrieved at 100% compression quality factor even though information is lost during the process. When the compression quality factor decreases further, the comparator is unable to retrieve the identical watermark image as too much information is being lost during the compression which can be determine by the file size after compression in Table 4.1.

### 4.5.2 Graphic Interchange Format (GIF)

GIF is a common format for image files, especially suitable for images containing large areas of the same colour. GIF format files of simple images are often smaller than the same file would be if stored in JPEG format, but GIF format does not store photographic images as well as JPEG.

As GIF files contain a maximum of 256 colors, this file format is ideal for simple graphics with minimal shading or color variation. Other types of graphics are better suited for the JPEG file format.

As JPEG compression does not work especially well with hard edges and lines in graphics images. Simple line drawings and pictures with transparent areas should be compressed into GIF rather than JPEG files.

In the LSB watermarking technique, the watermarked bitmap test image is converted to the GIF format using a image processing tool. The tool that is chosen for this project is IrfanView. Manual conversion of the GIF for compression and decompression is necessary as this format is more supported by the MATLAB *imread* and *imwrite* command. Therefore the bitmap watermarked image has to be converted manually before the decoding of the watermark.

In Table 4.2, the file size after the compression process is determined. The compression ratio is being calculated using the data. The data in this table is the same for the single watermark and multiple watermarks embedding scheme. It can be seen that after the compression, the watermark image retrieved is still identical to the original image.

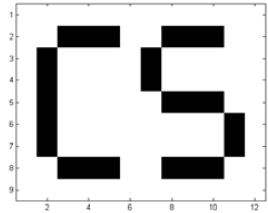
<b>Original bitmap test image size = 65kbps</b>			<b>Retrieved Watermark Image</b> 
<b>Image</b>	<b>File size (kbps)</b>	<b>Compression Ratio (%)</b>	
<b>Bird</b>	<b>46.4</b>	<b>71.38</b>	
<b>Lena</b>	<b>59.4</b>	<b>91.38</b>	
<b>Clock</b>	<b>49.9</b>	<b>76.77</b>	
<b>Bridge</b>	<b>60.4</b>	<b>92.92</b>	
<b>Camera</b>	<b>54.1</b>	<b>83.23</b>	

Table 4.2: GIF compression quality table for single and multiple watermarks



### 4.5.3 Improving the Basic Algorithm

During the initial stage of designing the LSB watermarking technique, only a single watermarked is embedded. After the embedding and decoding algorithms are designed and programmed using MATLAB, different bitmap test images are used to test the effectiveness of the program. However, after several round of testing, an identical watermark could not be retrieved as too much information is lost after subjected to compression.

Therefore a better solution is to embed multiple watermarks into the bitmap test images. This will allow a higher chance of retrieving the watermark image that is identical to the original watermark. To further enhance the multiple watermarks method, a comparator is designed. The comparator will determine the final watermark pixels value at individual pixel position.

Since only monochrome images are used, the pixels are normalized during the embedding stage. There will only be 2 level of pixels value. A '0' to represent black and '1' for white. Therefore the comparator will determine which is the most occurring value at that pixel position and make it the final pixel value.

## Chapter 5

# Conclusion and Future Work

The main advantage of the LSB watermarking technique is its high payload, whereas the main disadvantage lies in its low robustness, due to the fact that random changes destroy the coded watermark.

The LSB watermarking technique hosts some drawbacks due to its simplicity. The watermark will be corrupted with any addition of noise or lossy compression. An even better attack would be to simply set the LSB bits of each pixels to one and the watermark will be fully defeated with negligible impact on the bitmap test image. Furthermore, once the algorithm is discovered, the embedded watermark could be easily modified by an intermediate party.

An improvement is done by using a pseudo-random number generator as a secret key to determine the pixels to be embedded with the watermark in this project. Security of the watermark would be improved as the watermark could no longer be easily viewed by intermediate parties. The algorithm however would still be vulnerable to replacing the LSB's with a constant. Even in locations that were not used for watermarking bits, the impact of the substitution on the cover image

would be negligible. LSB watermarking is not robust at all but it is nevertheless important for a number of applications.

LSB modification proves to be a simple and fairly powerful tool for steganography. The characteristics of the LSB methods limit their applicability to steganographic scenarios and required a digital environment.

## **5.1 Achievement of Project Objectives**

The following objectives have been addressed:

1. Research the background information relating to watermarking and other information hiding techniques

Chapter 3.1 describes the information hiding techniques such as encryption, cryptography and steganography and Chapter 3.2 3.3 describes the watermarking relating to still images and audio.

2. Research the possible application areas of digital watermarking

Chapter 3.2.4 explained the different application of digital watermarking in relation to different areas.

3. Investigate several different watermarking algorithms

Chapter 3.2.5 describes the spread spectrum watermarking, patchwork watermarking, least significant bit watermarking and quantization index modulation watermarking techniques.

4. Implement one or more watermarking techniques and experimentally investigate the ability to recover the watermark when subjected to compression/decompression using JPEG and GIF encoding

Chapter 4 provide a more indepth explanation of the least significant bit watermarking technique that is designed and implemented in this project. The test results when the watermark is subjected to JPEG and GIF compression is shown in Chapter 4.5 and the detail test results in Appendix B.

5. Investigate methods to improve the robustness of the watermark recovery when the image is subjected to lossy compression

The multiple watermarks and comparator is designed to improve the robustness of the retrieved watermark. The test results are shown in Chapter 4.5 and Appendix B.

## 5.2 Further Work

The LSB watermarking technique is designed and implemented in this project. The JPEG and GIF compression are used to test the robustness of the watermarking technique. However, from the test results in Chapter 4.5 and Appendix B, the LSB watermarking technique is not robust enough when the compression factor gets smaller. Therefore some of the following can be implemented as a further work for this project.

- To implement the watermarking on RGB images.
- To design and implement other watermarking techniques.

- To implement spatial watermarking techniques.
- To test other compression format such as TIFF, PNG and the new JPEG2000.
- To implement the LSB watermarking technique algorithm design in this project in other digital media content such as audio and video.

# References

- abd R. Akalu, R. O. (2004), ‘Legal policy and digital rights management’, *Proceedings of the IEEE* **92**(6), 997–1003.
- B. Macq, J. D. & Delp, E. J. (2004), ‘Benchmarking of image watermarkign algorithms for digital rights management’, *Proceedings of the IEEE* **92**(6), 971–983.
- Bleumer, G. (2004), *Watermarking*.  
<http://www.win.tue.nl/~henkvt/GB1.Watermarking.pdf>  
current April 2004.
- Chen, B. & Wornell, G. W. (2001), ‘Quantization index modulation: A class of probably good methods for digital watermarking and information embedding’, *IEEE Transactions on Information Theory* **47**(4).
- Cole, E. (2003), *Hiding in Plain Sight*, Wiley, chapter 2 - 4.
- D. Kundar, C.-Y. Lin, B. M. & H.Yu (2004), ‘Scanning the special issue on enabling security technologies for digital rights’, *Proceedings of the IEEE* **92**(6), 879–882+.
- de C.T. Gromes, L. (2003), ‘Audio watermarking and fingerprinting: For which application?’, *Journal of New Music Research* **32**(1), 65–81.
- I. J. Cox, M. L. M. & Bloom, J. A. (2000), ‘Watermarking applications and their properties’, *Proceedings of the IEEE International Conference of Information Technology: Coding and Computing* pp. 6–10.

- I. J. Cox, M. L. M. & Bloom, J. A. (2001), *Digital Watermarking*, Morgan Kaufmann, chapter 2.
- Kalker, A. (1998), *Security Risk for Publicly Available Watermark Detectors (A)*, *Benelux Information Theory Symposium*, Velhoven, The Netherlands.  
<http://www.jjtc.com/Steganography/bib/3000123.htm>  
current Feb 2004.
- Katzenbeisser, S. (1999), *Principle in Steganography in Information Techniques for Steganography and Digital Watermarking*, Artec House, Reading, Northwood, chapter 1, pp. 2–40.
- Kutter, M. (2001), *Digital Watermarking Frequently Asked Questions*.  
<http://www.watermarkingworld.org/faq.html>  
current March 2004.
- Lyon, G. A. (2002), ‘A quick-reference list of organizations and standards for digital rights management’.
- Mao, W. (2003), *Modern Cryptography: Theory and Practice*, Prentice Hall PTR, chapter 7-8.
- Pitas, I. (1997), *Digital Watermarks: For Copyright Protection of Still Images, audio and video.*, University of Nowhere.  
<http://poseidon.csd.auth.gr/signatures/>  
current March 2004.
- Provos, N. (1999), *Outguess: What is Steganography?*  
<http://www.outguess.com>  
current April 2004.
- Rob H. Koenen, J. Lacy, M. M. & Mitchell, S. (2004), ‘The long march to interoperable digital rights management’, *Proceedings of the IEEE* **92**(6), 883–897.
- Watermarks: Protecting the Image* (2004).

---

[http://www.research.ibm.com/image\\_apps/watermark.html](http://www.research.ibm.com/image_apps/watermark.html)  
current May 2003.

Westphal, K. (2003), *Steganography Revealed*.

<http://www.securityfocus.com/infocus/1684>  
current April 2004.



# Appendix A

## Project Specification

University of Southern Queensland  
FACULTY OF ENGINEERING AND SURVEYING

**ENG 4111/4112 Research Project**  
**PROJECT SPECIFICATION**

FOR: LOH SHIAU PING

TOPIC: Watermarking for securing digital media content

SUPERVISORS: DR JOHN LEIS

ENROLMENT: ENG 4111 – S1, X, 2004;  
ENG 4112 – S2, X, 2004

PROJECT AIM: The aim of this project is to experimentally evaluate the effect of compression on embedded watermarks in digital media. Current research in the area of image and audio watermarking is to be investigated, and the robustness of simple watermarking methods to lossy compression should be experimentally evaluated using a suitable software platform.

PROGRAMME: Issue B, 24 September 2004

1. Research the background information relating to watermarking and other information hiding techniques.
2. Research the possible application areas of digital watermarking.
3. Investigate several different watermarking algorithms.
4. Implement one or more watermarking techniques and experimentally investigate the ability to recover the watermark when subjected to compression/decompression using JPEG and GIF encoding.
5. Investigate methods to improve the robustness of the watermark recovery when the image is subjected to lossy compression.

# Appendix B

## Detail Test Results

## B.1 Single Watermark

### B.1.1 Bird

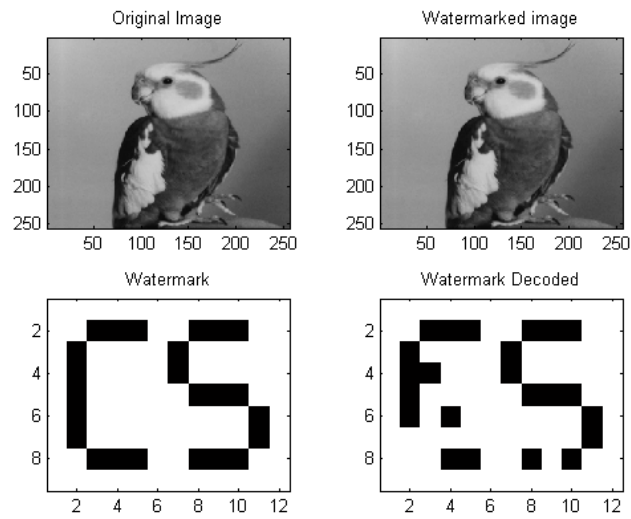


Figure B.1: Test result with 100% compression quality factor

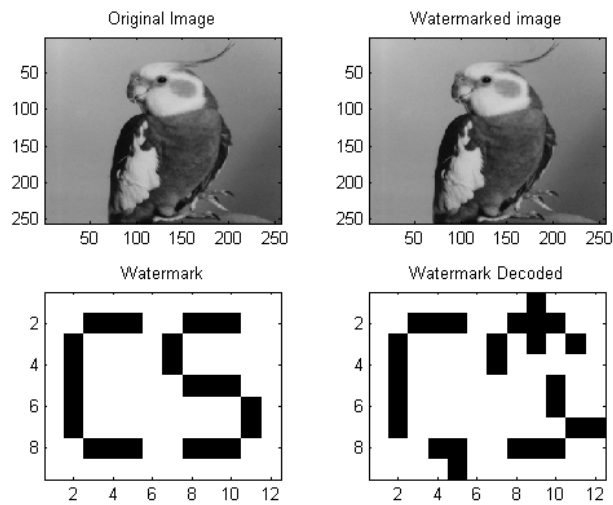


Figure B.2: Test result with 99% compression quality factor

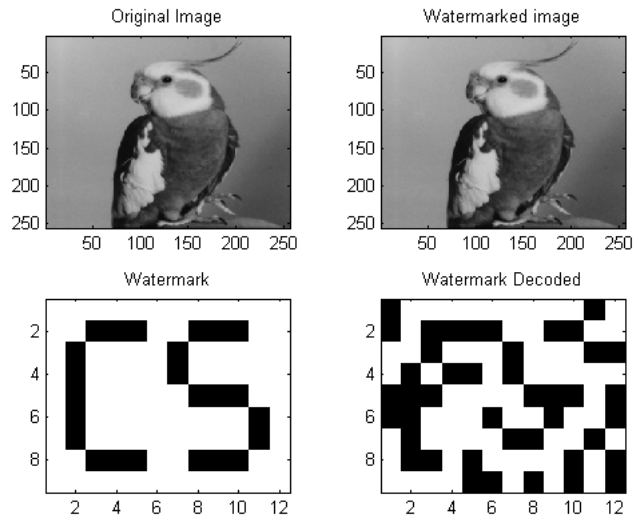


Figure B.3: Test result with 98% compression quality factor

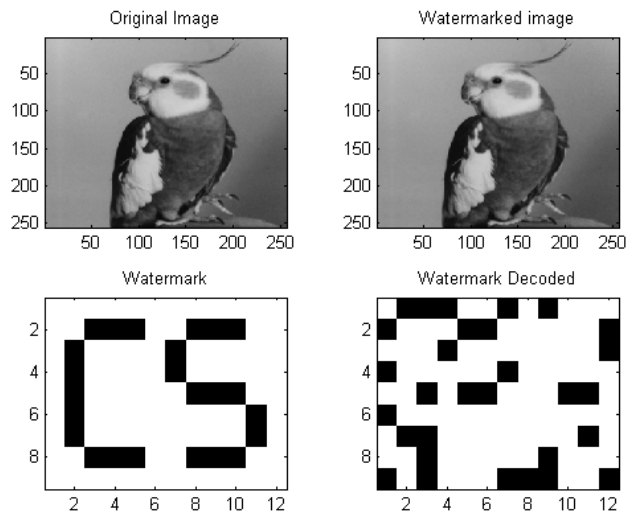


Figure B.4: Test result with 95% compression quality factor

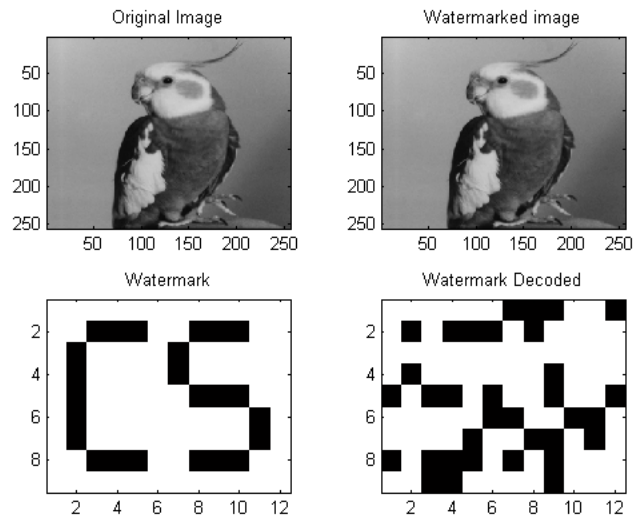


Figure B.5: Test result with 90% compression quality factor

### B.1.2 Lena

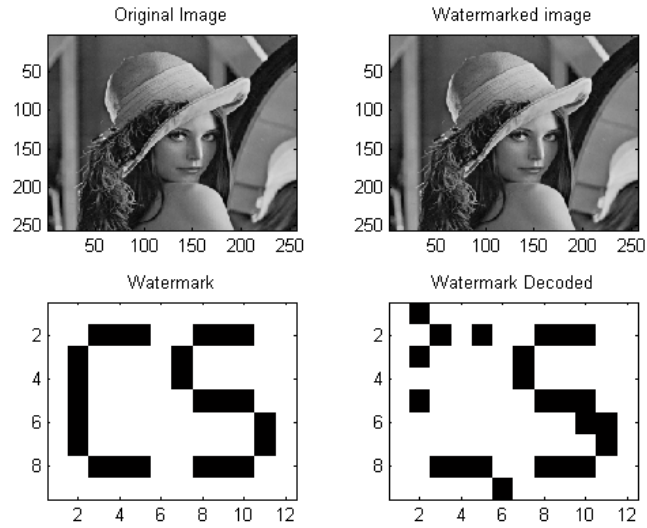


Figure B.6: Test result with 100% compression quality factor

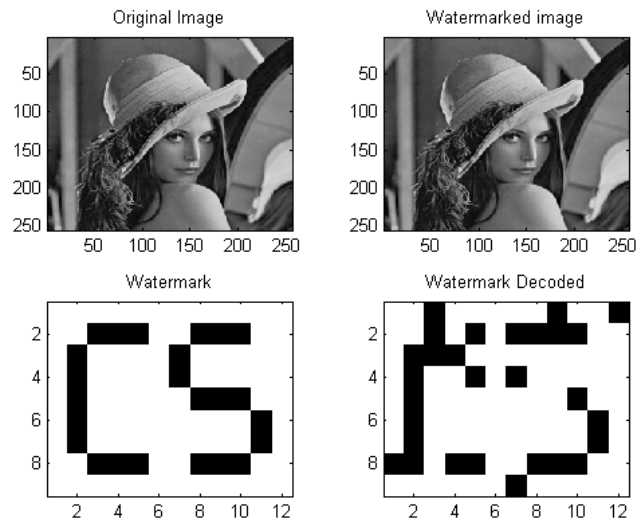


Figure B.7: Test result with 99% compression quality factor

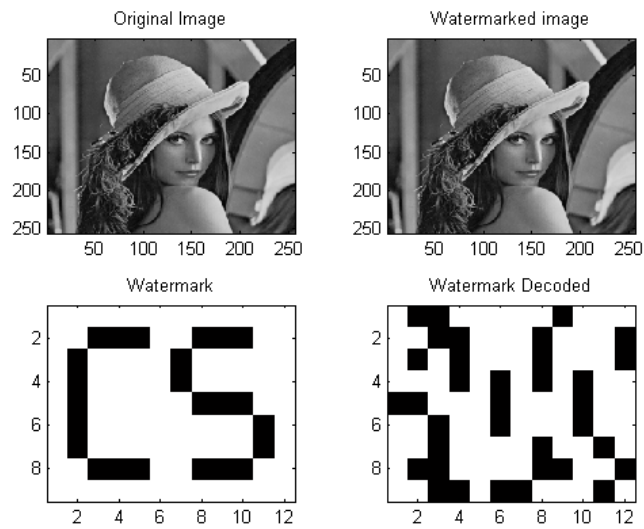


Figure B.8: Test result with 98% compression quality factor

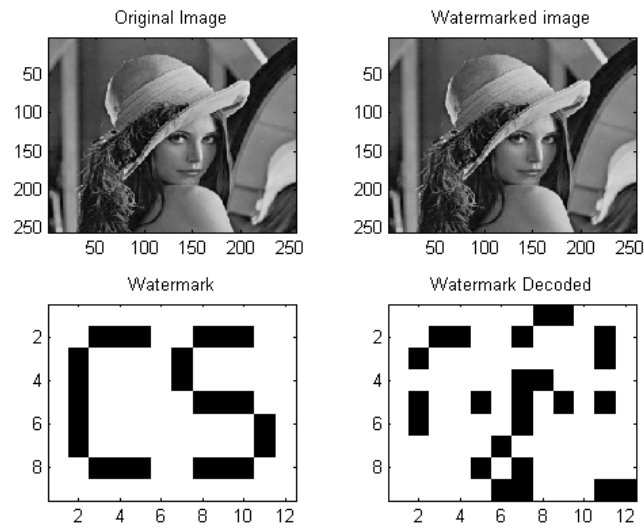


Figure B.9: Test result with 95% compression quality factor

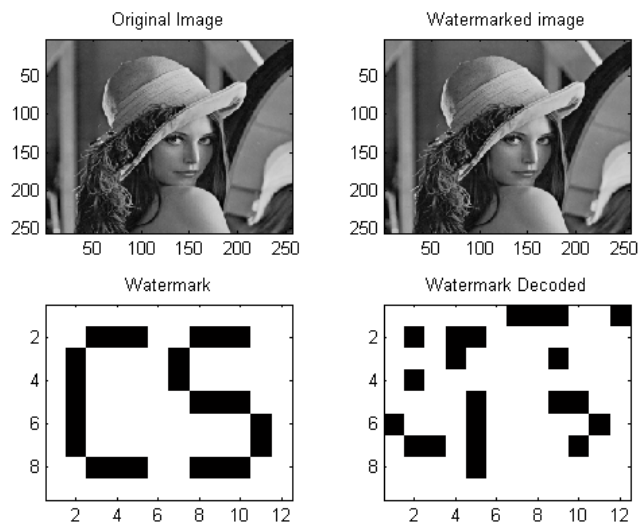


Figure B.10: Test result with 90% compression quality factor



## B.1.3 Clock

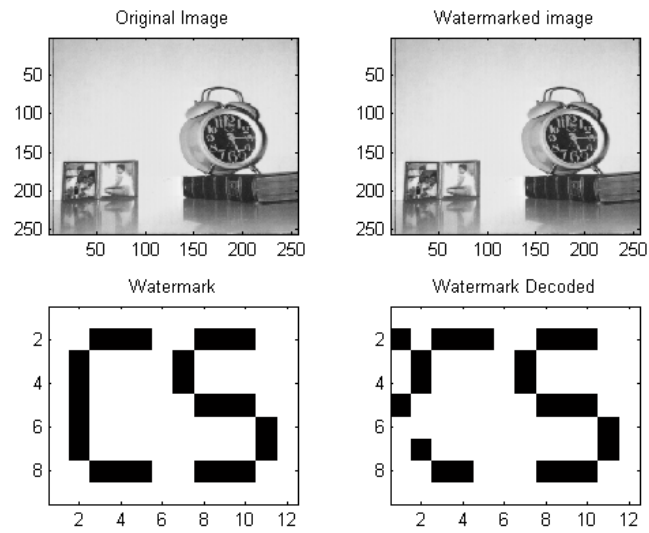


Figure B.11: Test result with 100% compression quality factor

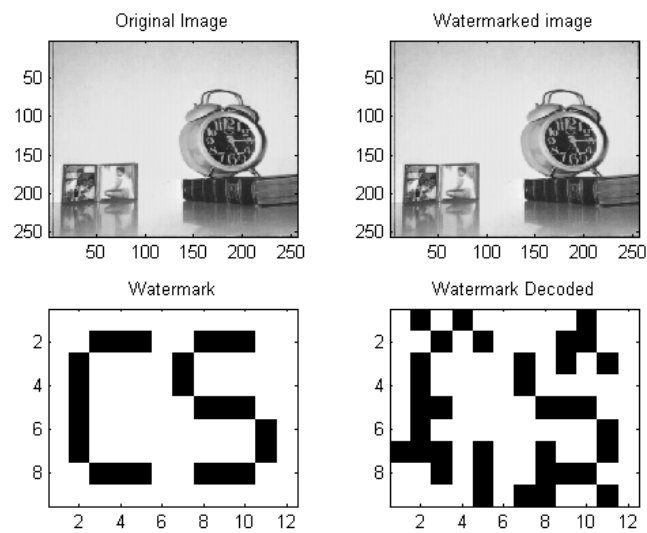


Figure B.12: Test result with 99% compression quality factor

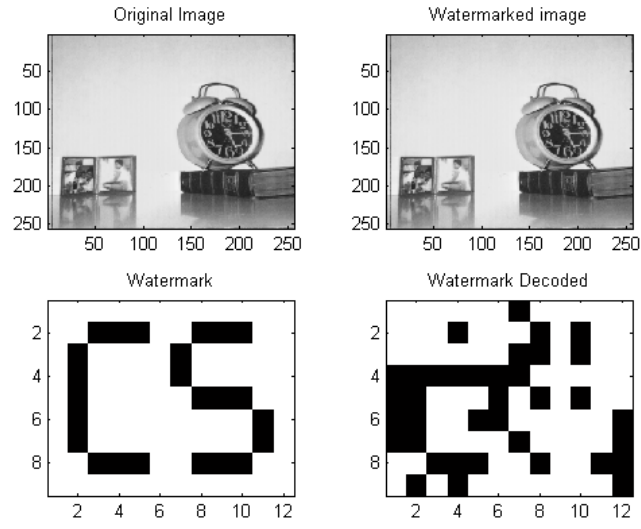


Figure B.13: Test result with 98% compression quality factor

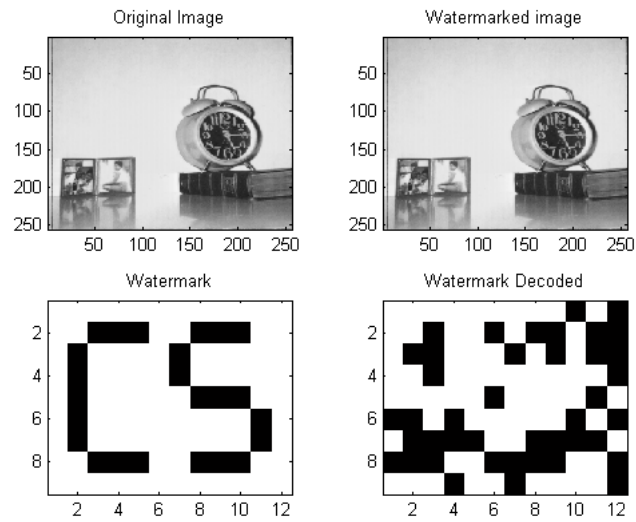


Figure B.14: Test result with 95% compression quality factor

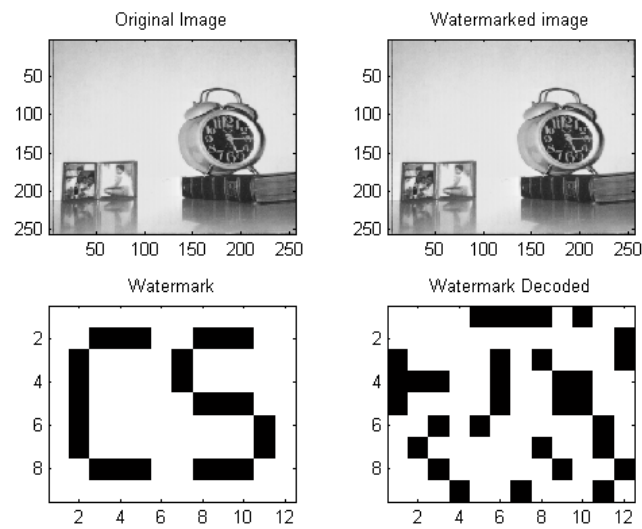


Figure B.15: Test result with 90% compression quality factor

#### B.1.4 Bridge

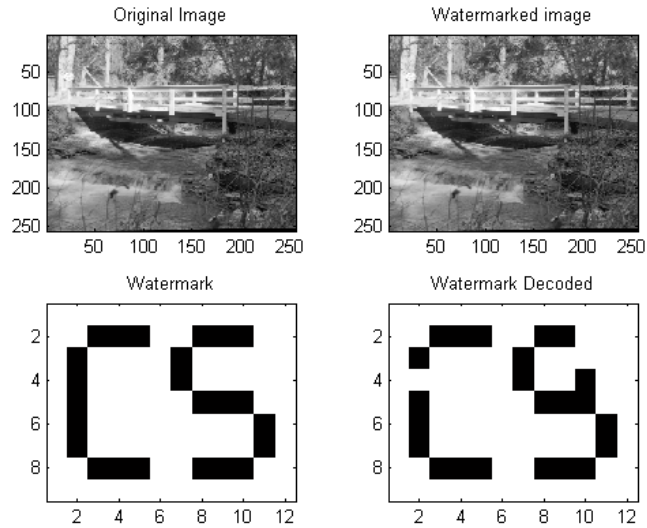


Figure B.16: Test result with 100% compression quality factor

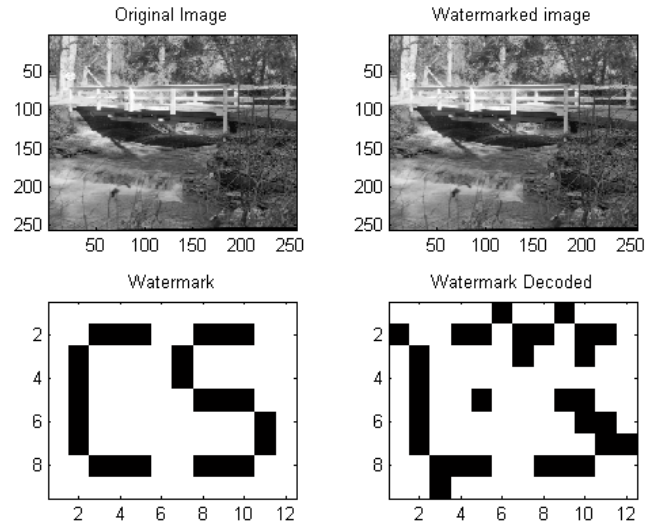


Figure B.17: Test result with 99% compression quality factor

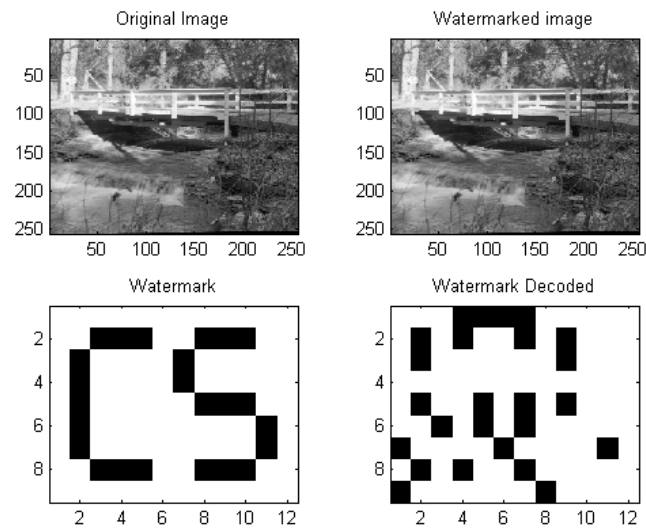


Figure B.18: Test result with 98% compression quality factor

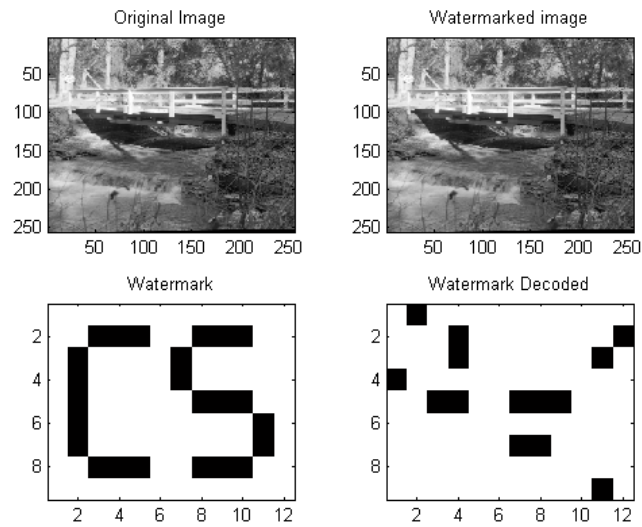


Figure B.19: Test result with 95% compression quality factor

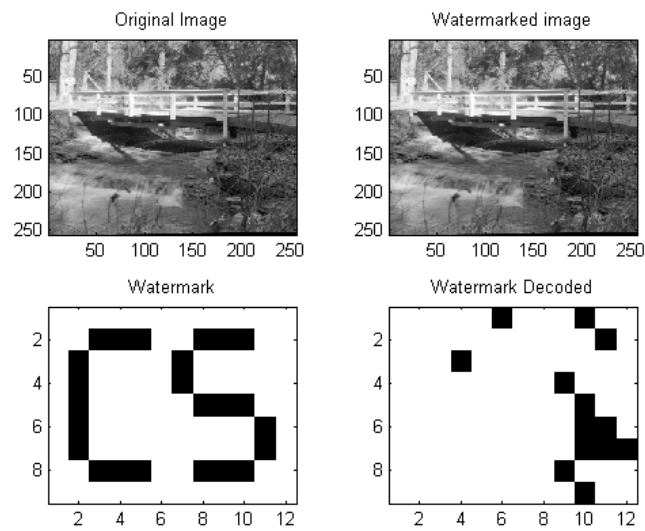


Figure B.20: Test result with 90% compression quality factor

## B.1.5 Camera

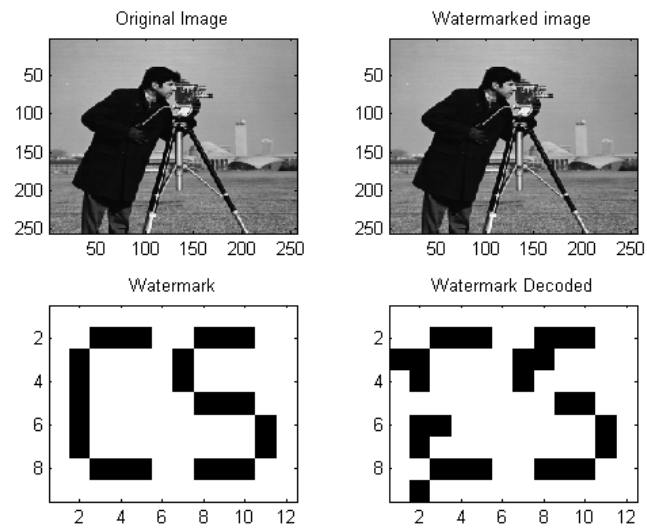


Figure B.21: Test result with 100% compression quality factor

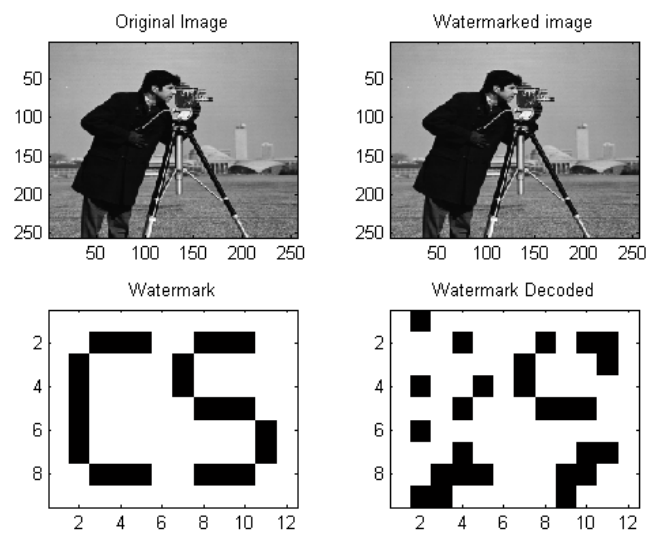


Figure B.22: Test result with 99% compression quality factor

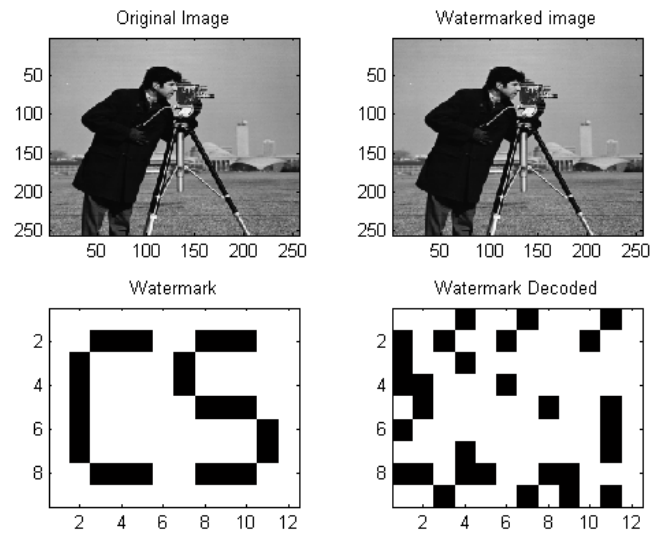


Figure B.23: Test result with 98% compression quality factor

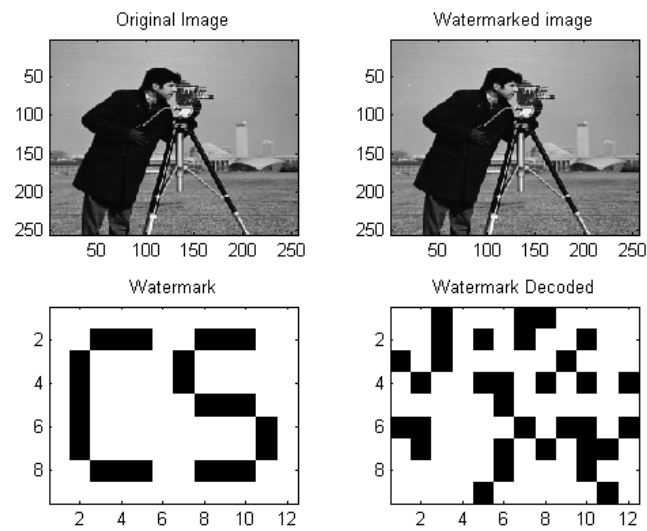


Figure B.24: Test result with 95% compression quality factor

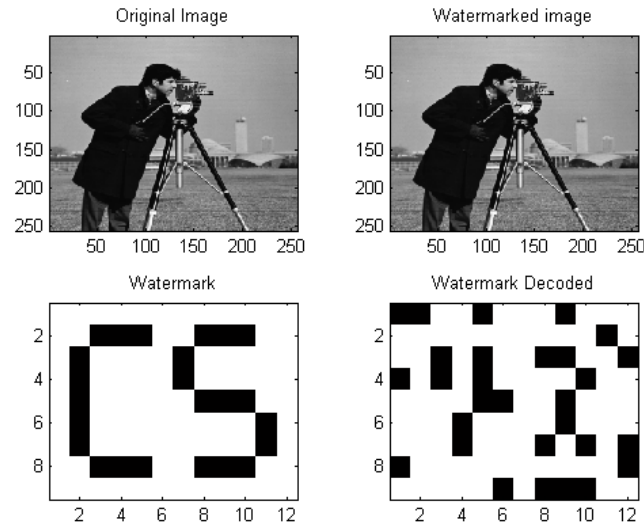


Figure B.25: Test result with 90% compression quality factor

## B.2 Multiple Watermarks without Comparator

### B.2.1 Bird

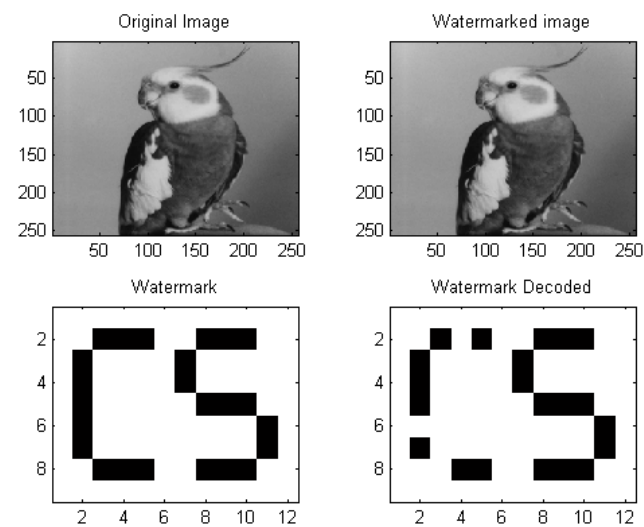


Figure B.26: Test result with 100% compression quality factor



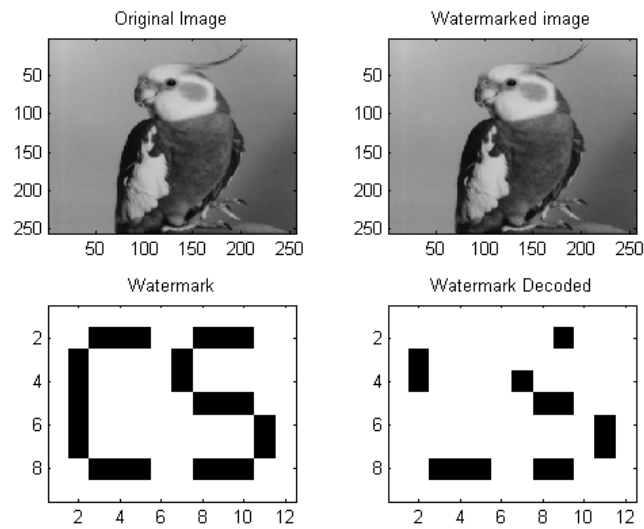


Figure B.27: Test result with 99% compression quality factor

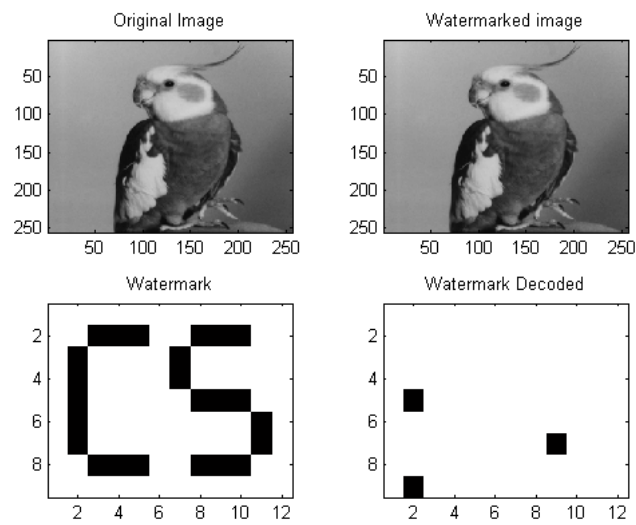


Figure B.28: Test result with 98% compression quality factor

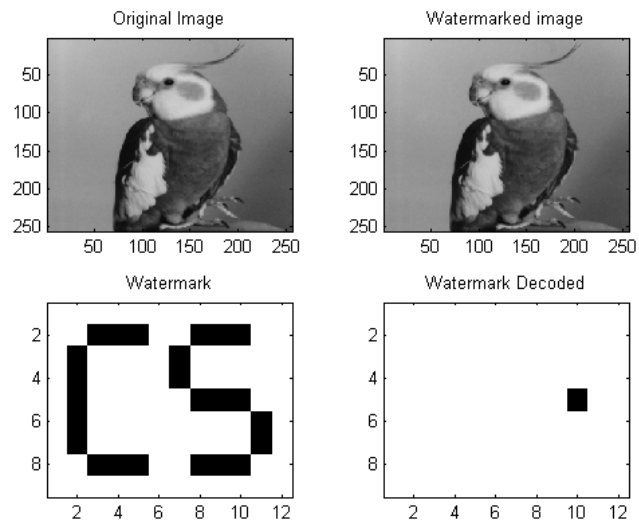


Figure B.29: Test result with 95% compression quality factor

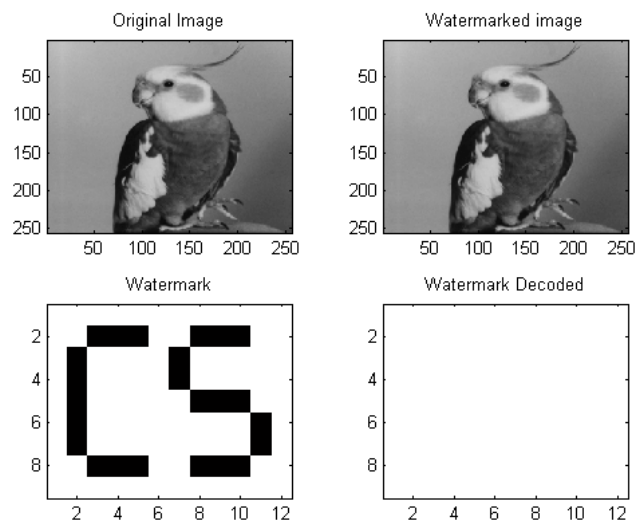


Figure B.30: Test result with 90% compression quality factor

## B.2.2 Lena

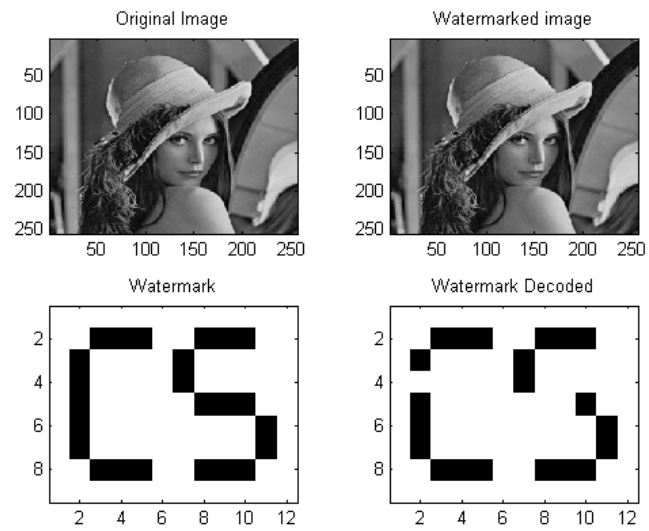


Figure B.31: Test result with 100% compression quality factor

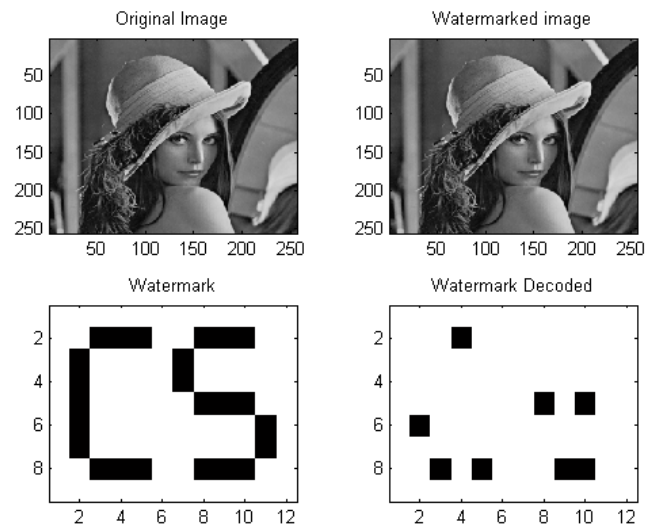


Figure B.32: Test result with 99% compression quality factor

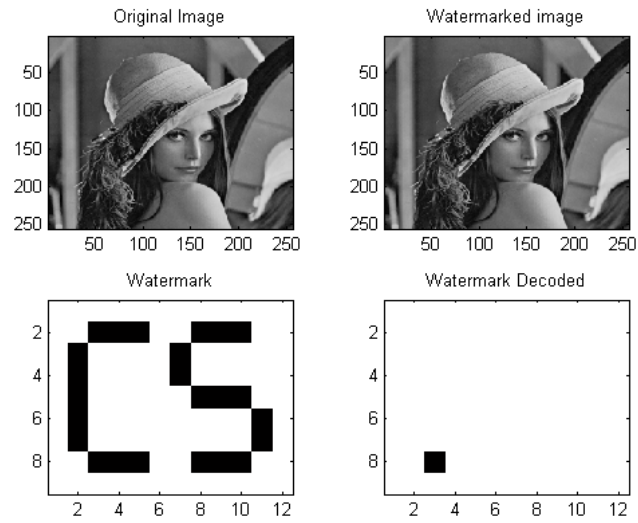


Figure B.33: Test result with 98% compression quality factor

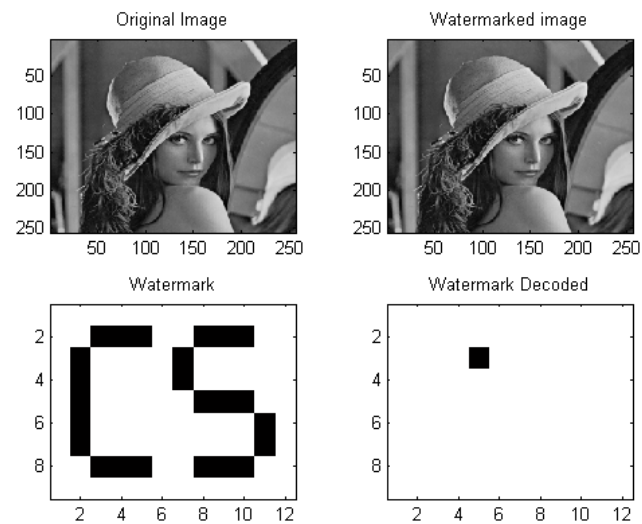


Figure B.34: Test result with 95% compression quality factor

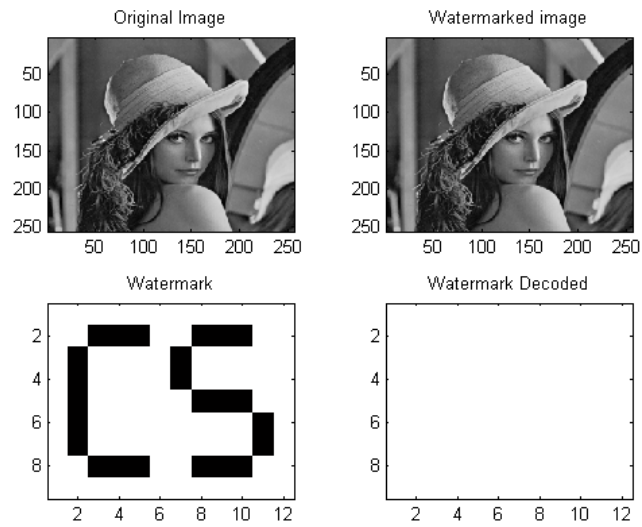


Figure B.35: Test result with 90% compression quality factor

### B.2.3 Clock

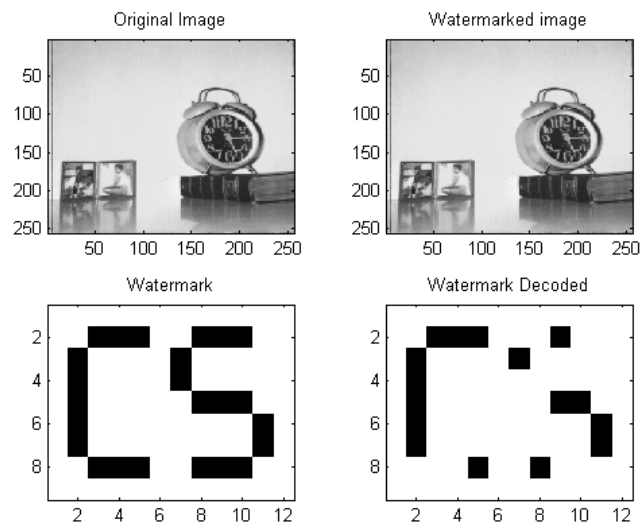


Figure B.36: Test result with 100% compression quality factor

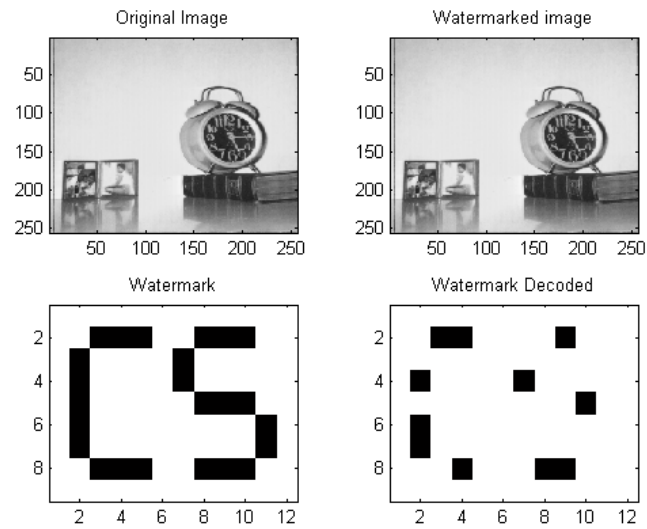


Figure B.37: Test result with 99% compression quality factor

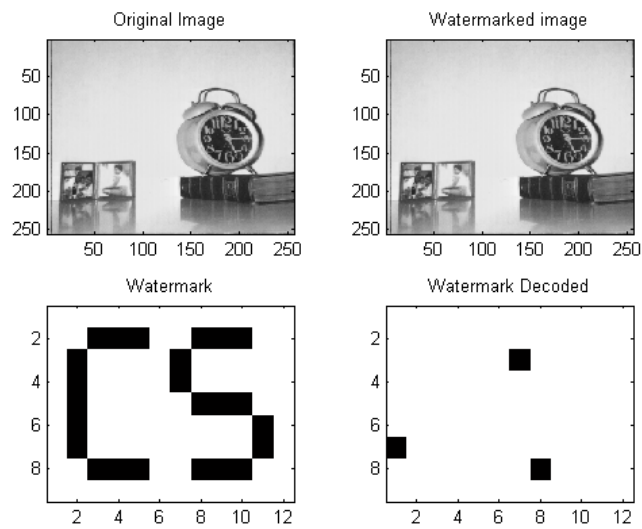


Figure B.38: Test result with 98% compression quality factor

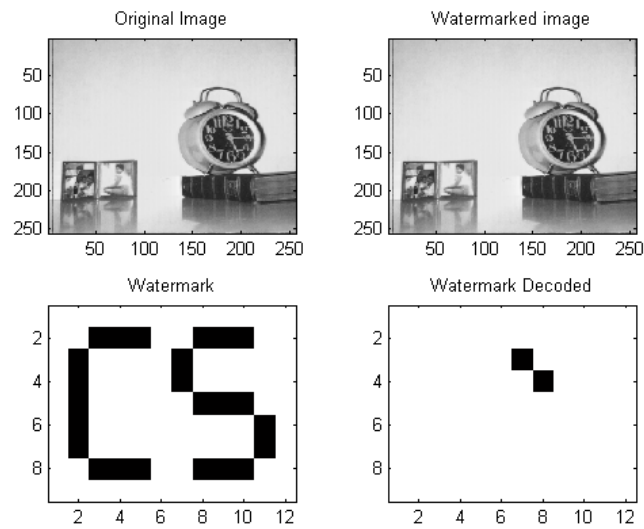


Figure B.39: Test result with 95% compression quality factor

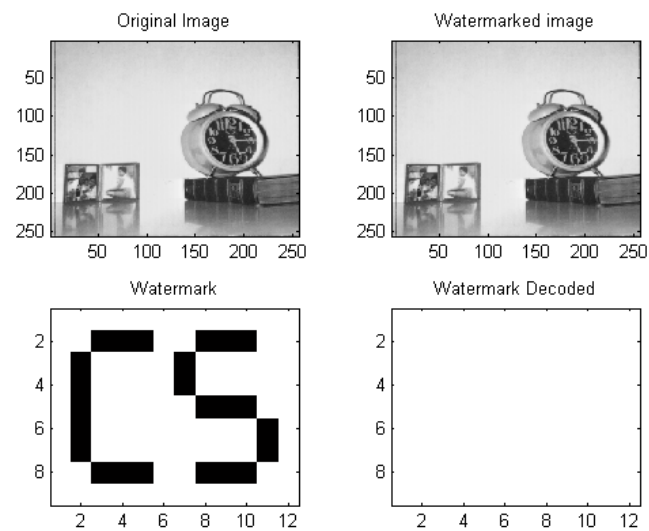


Figure B.40: Test result with 90% compression quality factor

## B.2.4 Bridge

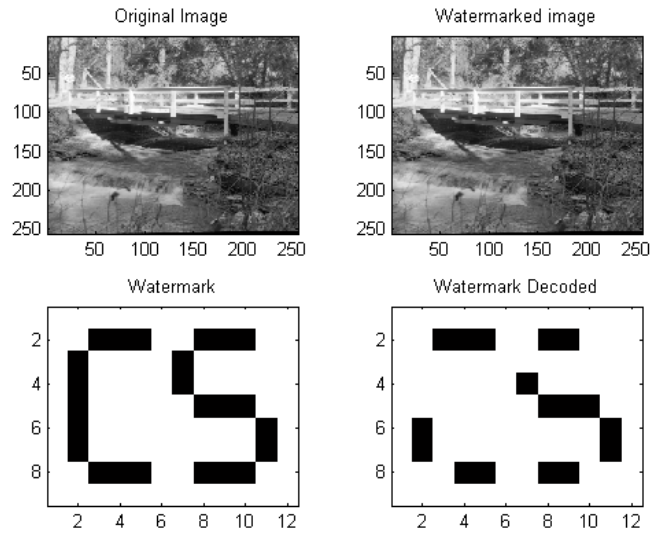


Figure B.41: Test result with 100% compression quality factor

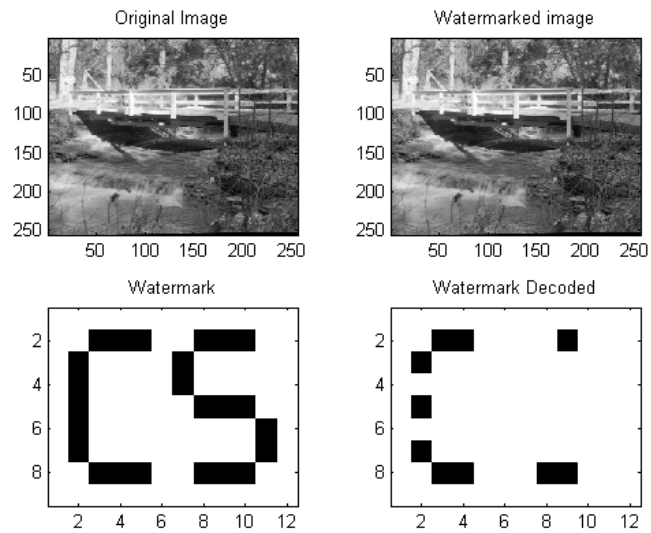


Figure B.42: Test result with 99% compression quality factor



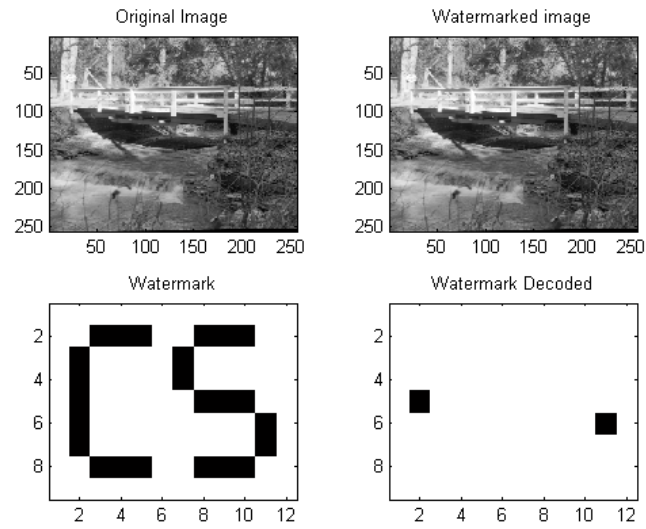


Figure B.43: Test result with 98% compression quality factor

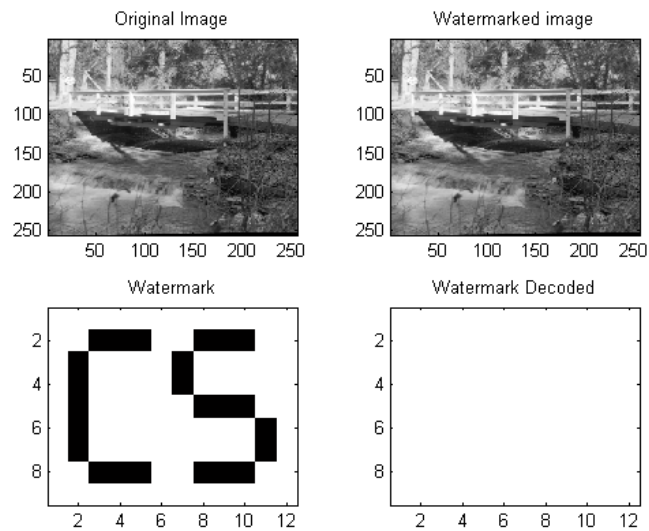


Figure B.44: Test result with 95% compression quality factor

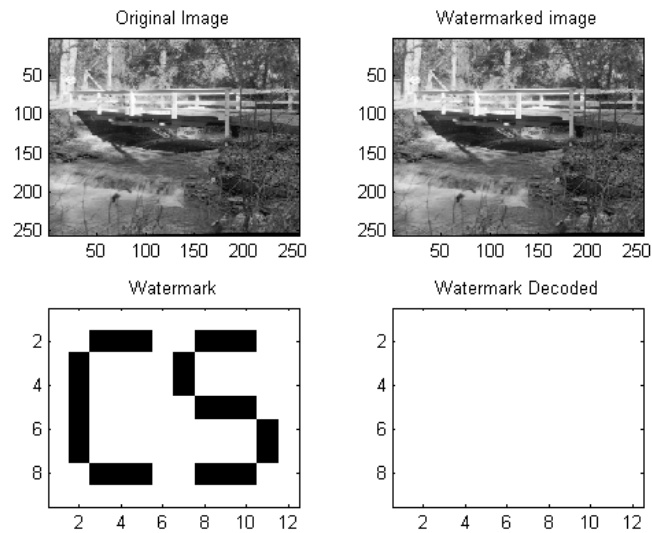


Figure B.45: Test result with 90% compression quality factor

### B.2.5 Camera

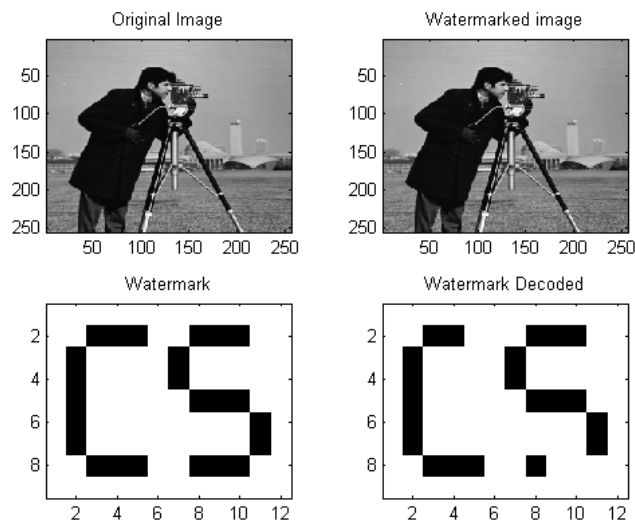


Figure B.46: Test result with 100% compression quality factor

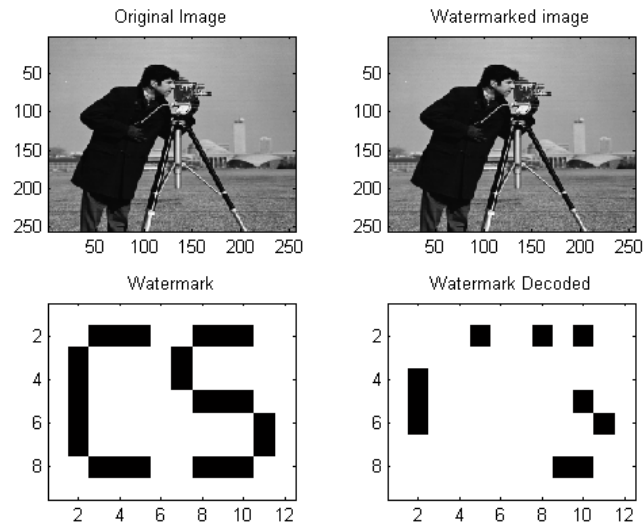


Figure B.47: Test result with 99% compression quality factor

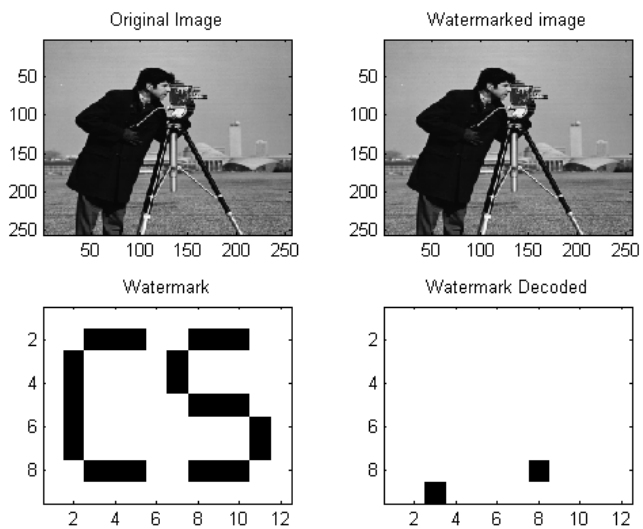


Figure B.48: Test result with 98% compression quality factor

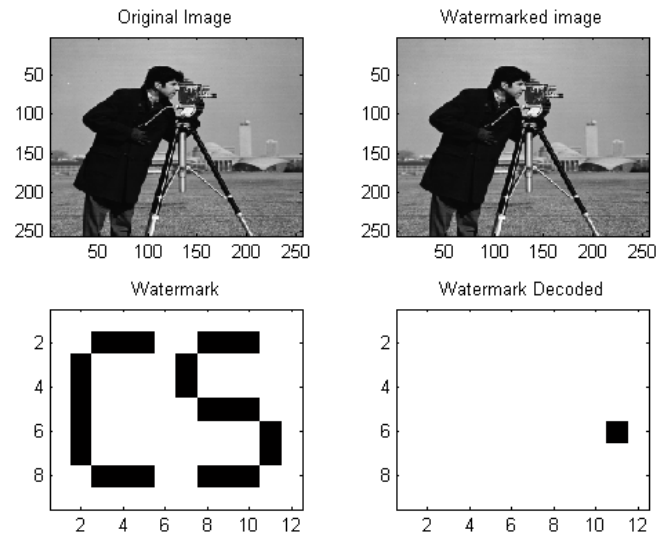


Figure B.49: Test result with 95% compression quality factor

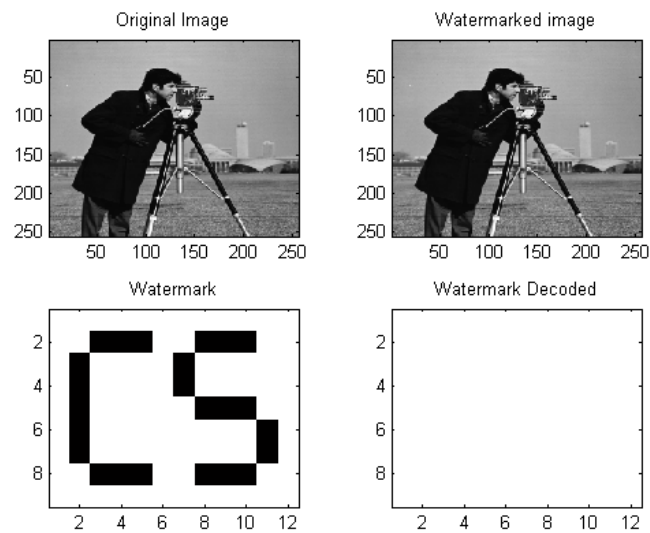


Figure B.50: Test result with 90% compression quality factor

## B.3 Multiple Watermark with Comparator

### B.3.1 Bird

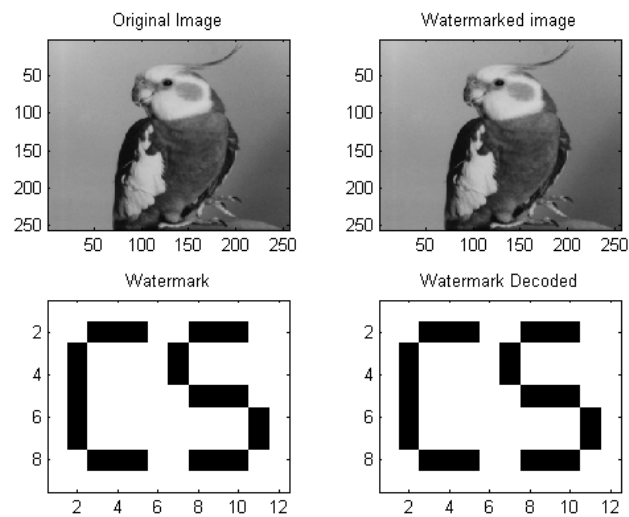


Figure B.51: Test result with 100% compression quality factor

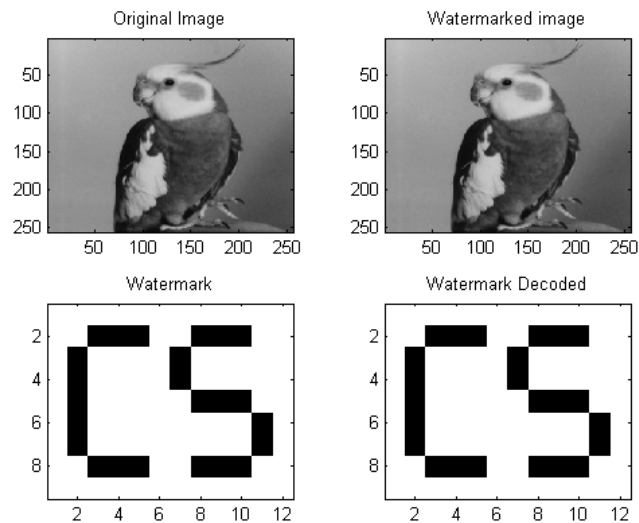


Figure B.52: Test result with 99% compression quality factor

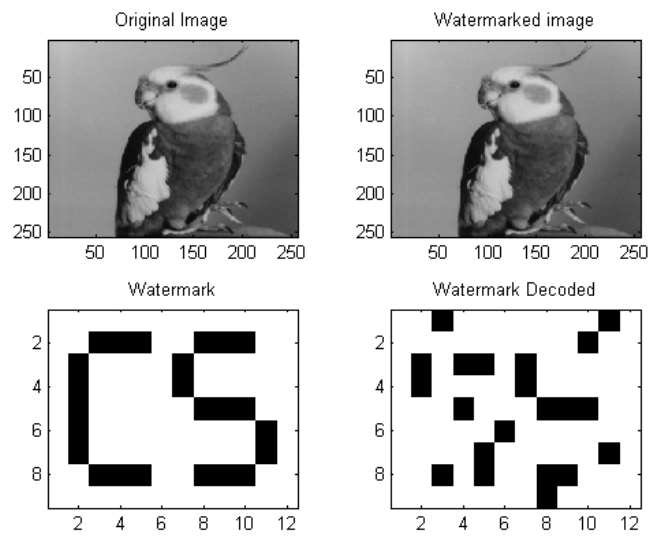


Figure B.53: Test result with 98% compression quality factor

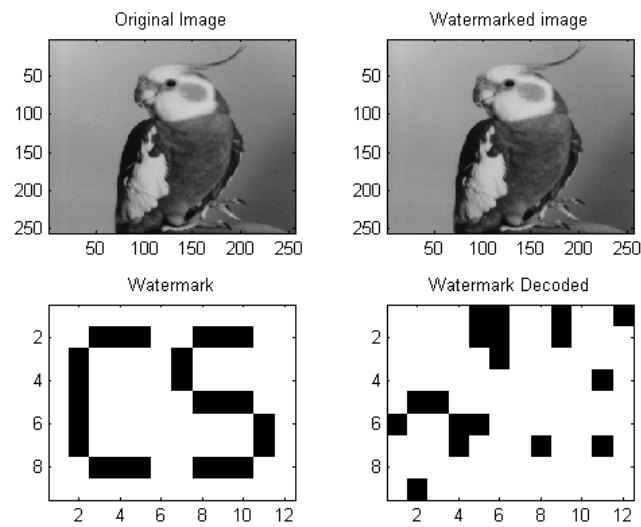


Figure B.54: Test result with 95% compression quality factor

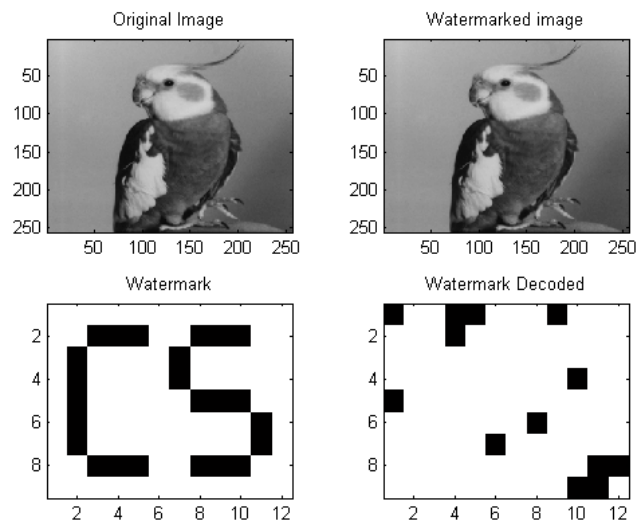


Figure B.55: Test result with 90% compression quality factor

### B.3.2 Lena

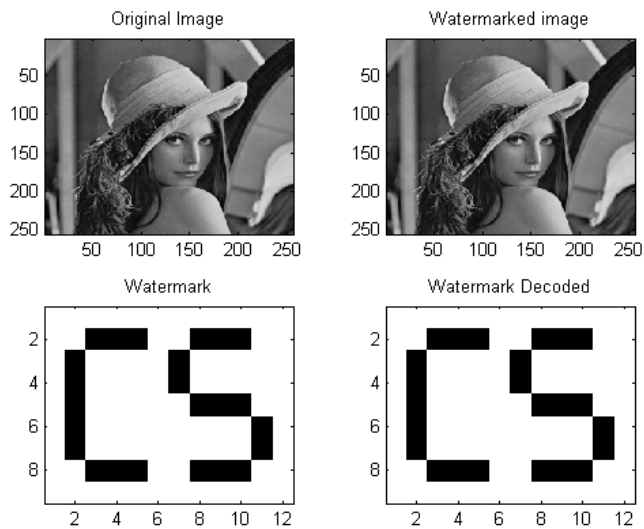


Figure B.56: Test result with 100% compression quality factor

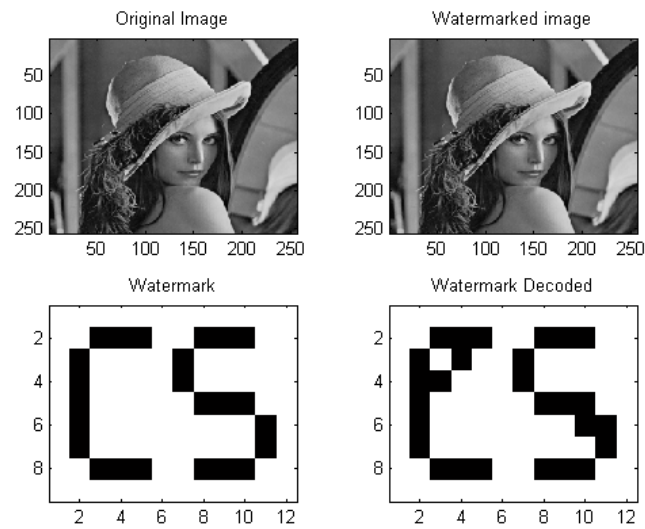


Figure B.57: Test result with 99% compression quality factor

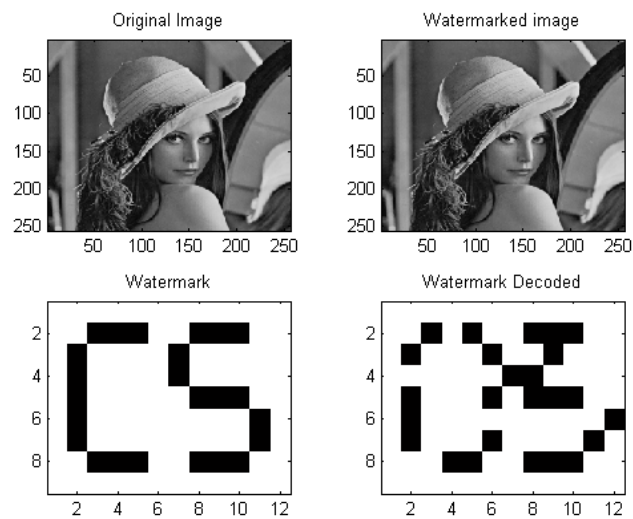


Figure B.58: Test result with 98% compression quality factor



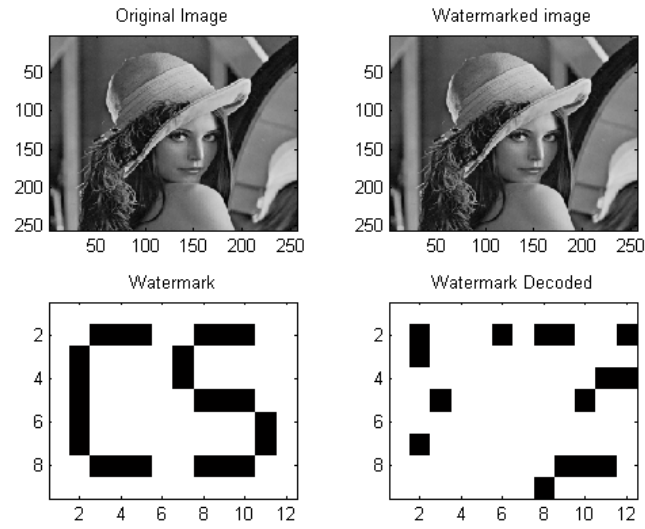


Figure B.59: Test result with 95% compression quality factor

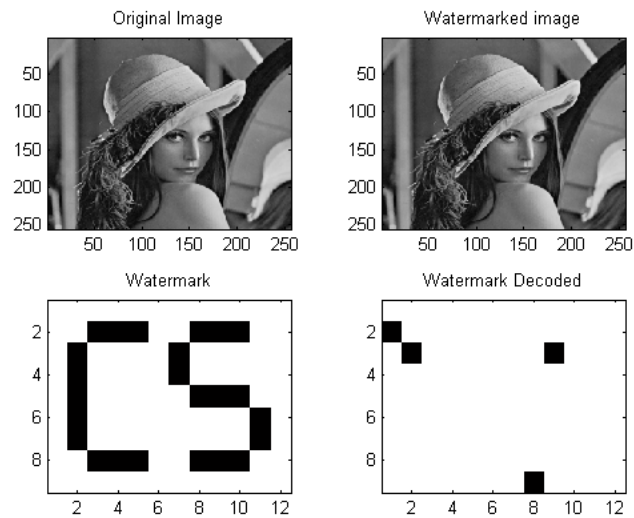


Figure B.60: Test result with 90% compression quality factor

## B.3.3 Clock

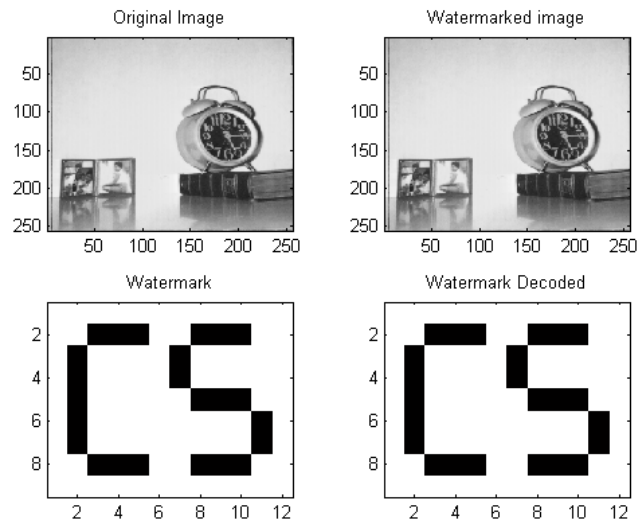


Figure B.61: Test result with 100% compression quality factor

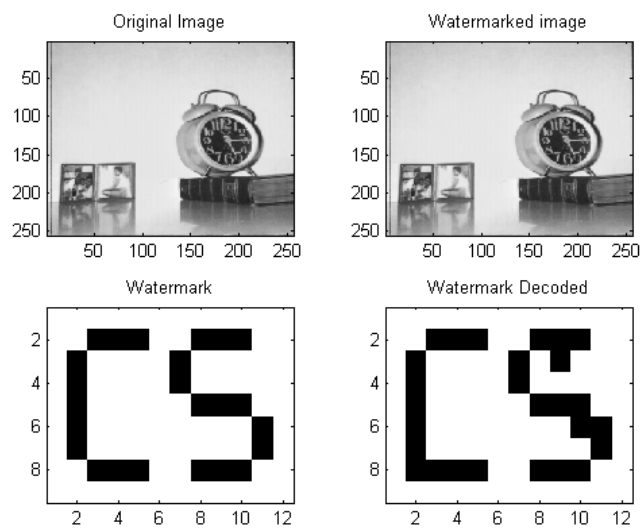


Figure B.62: Test result with 99% compression quality factor

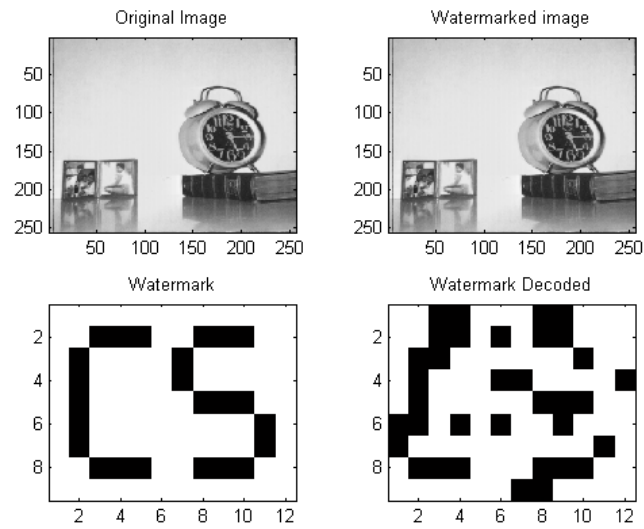


Figure B.63: Test result with 98% compression quality factor

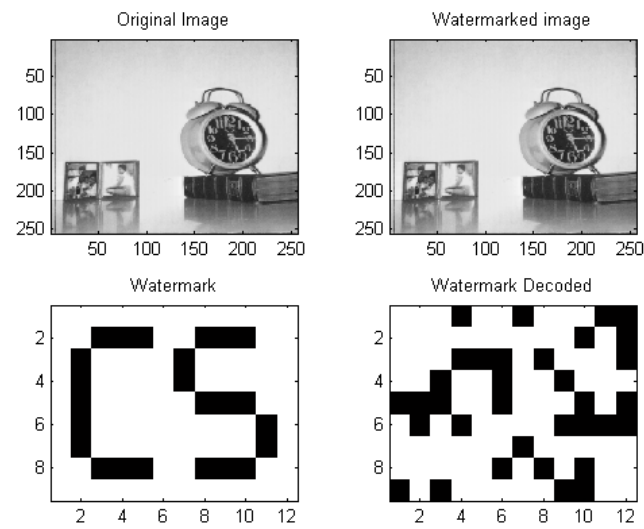


Figure B.64: Test result with 95% compression quality factor

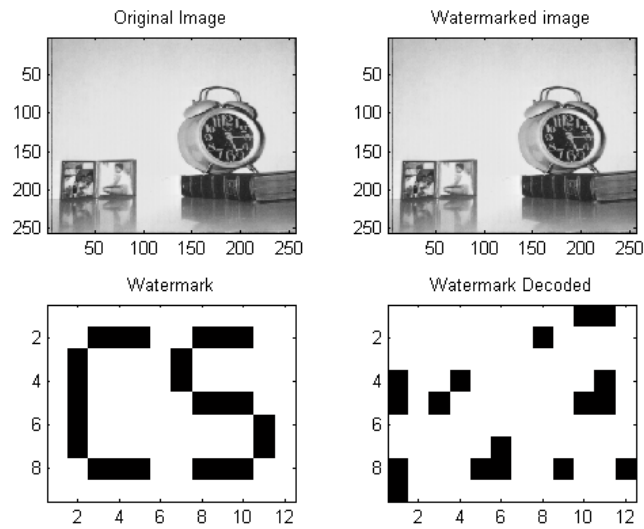


Figure B.65: Test result with 90% compression quality factor

### B.3.4 Bridge

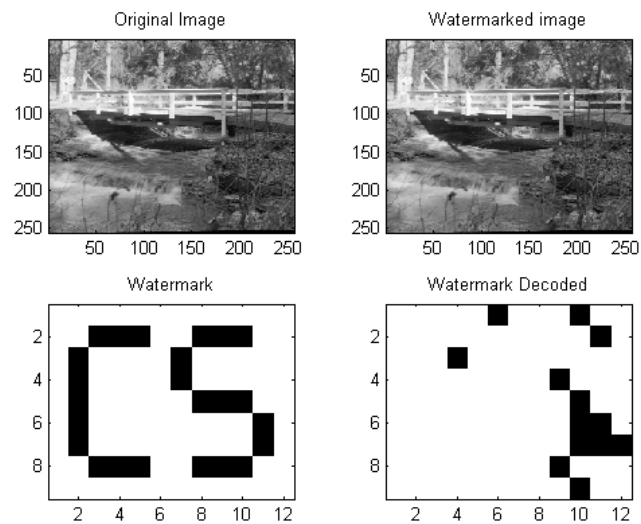


Figure B.66: Test result with 100% compression quality factor

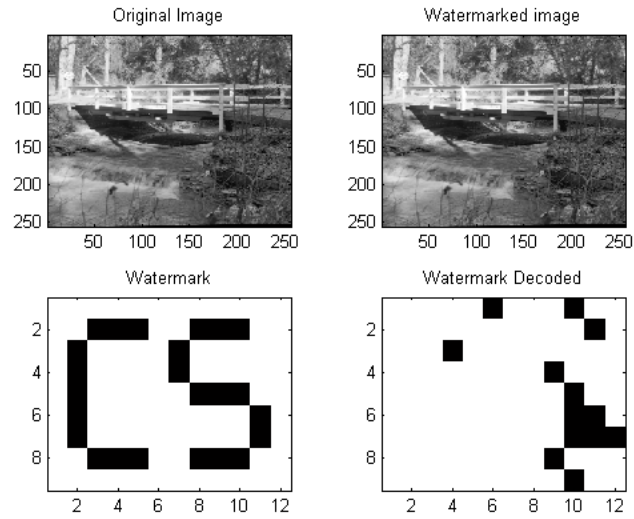


Figure B.67: Test result with 99% compression quality factor

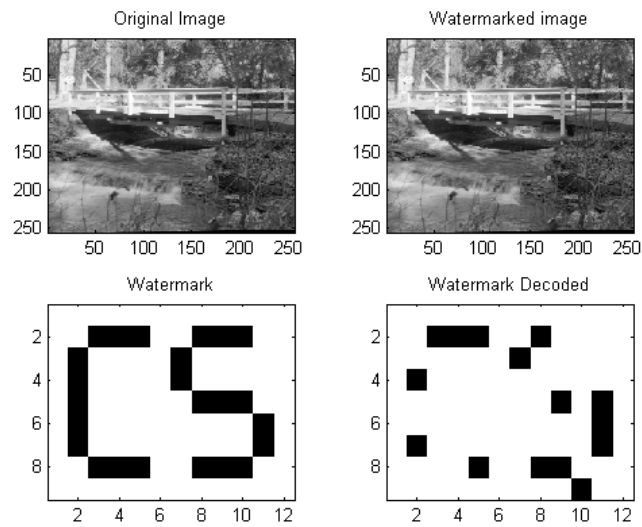


Figure B.68: Test result with 98% compression quality factor

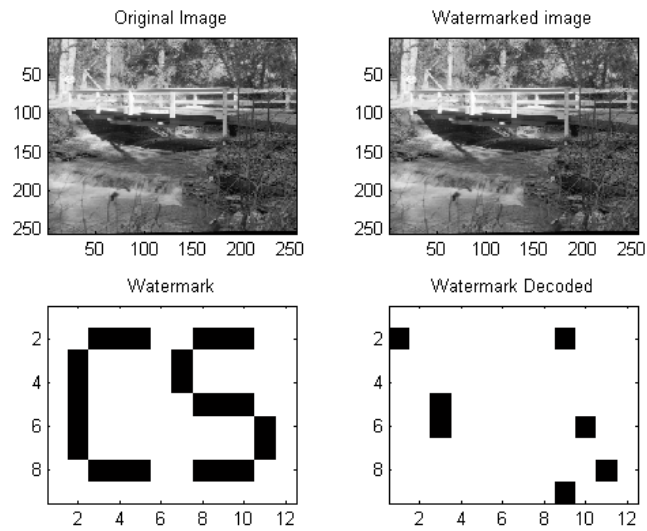


Figure B.69: Test result with 95% compression quality factor

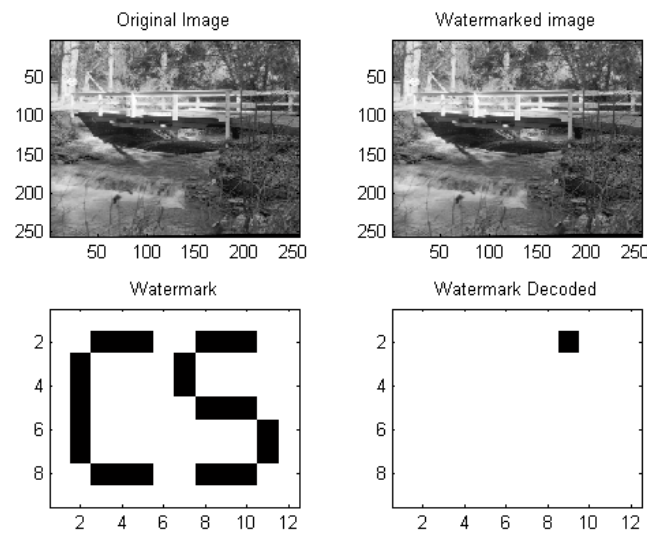


Figure B.70: Test result with 90% compression quality factor

## B.3.5 Camera

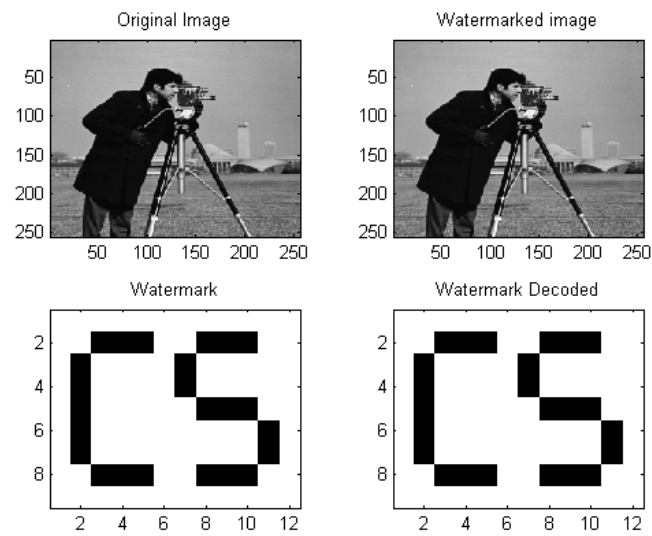


Figure B.71: Test result with 100% compression quality factor

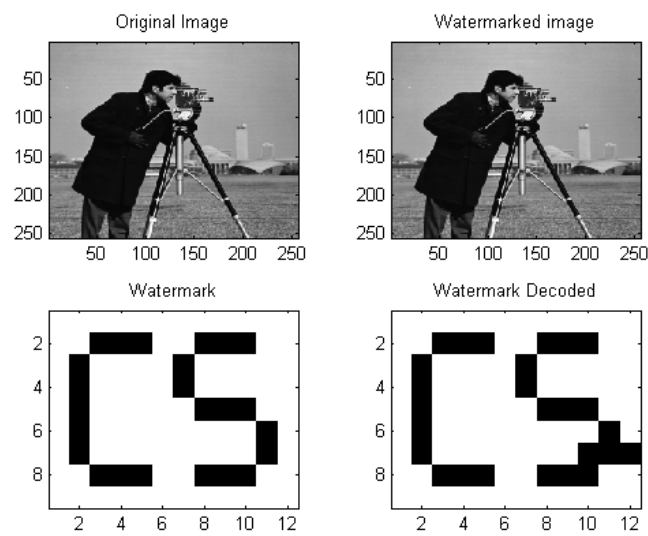


Figure B.72: Test result with 99% compression quality factor

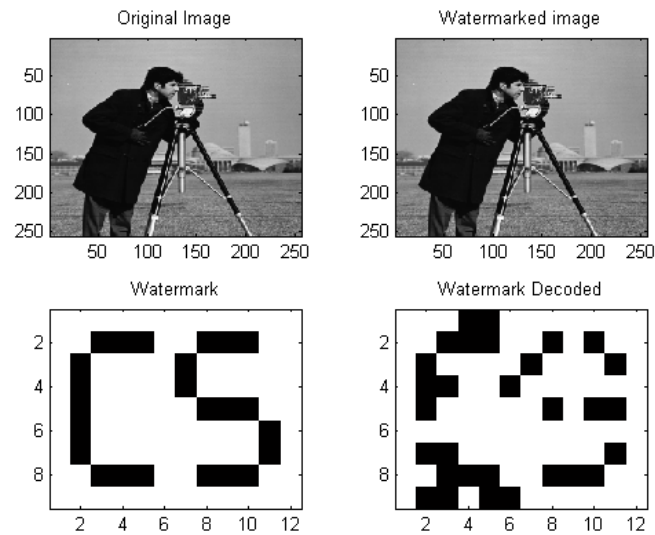


Figure B.73: Test result with 98% compression quality factor

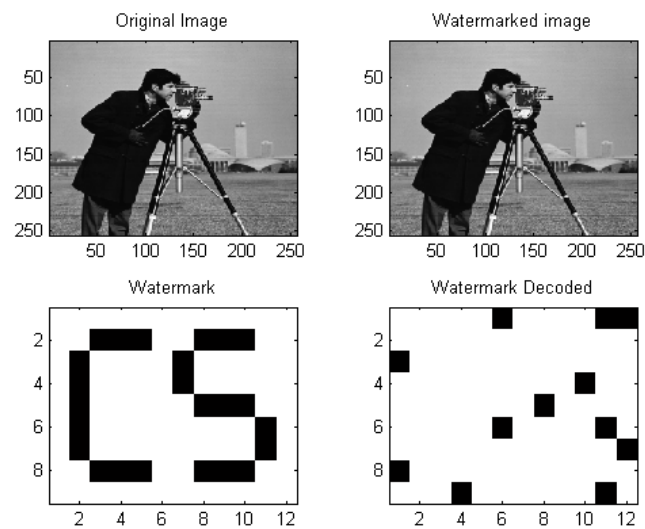


Figure B.74: Test result with 95% compression quality factor



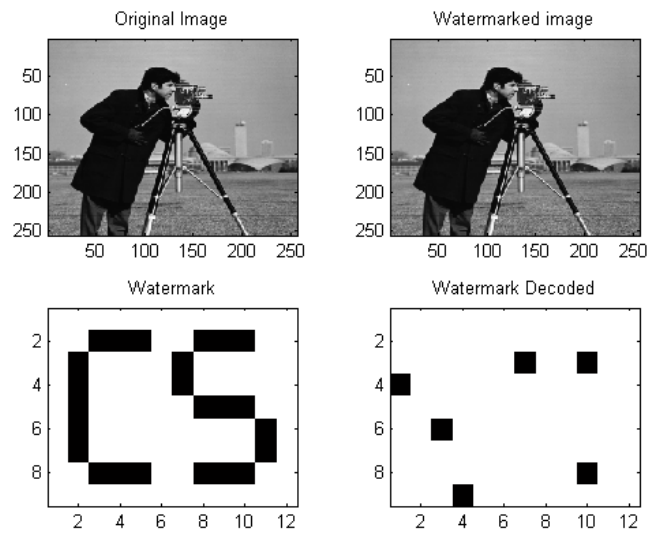


Figure B.75: Test result with 90% compression quality factor

# Appendix C

## Source Code

## C.1 Main.m

```
%Main Program
%This is the main program to run for the LSB watermarking technique.
%A menu will be shown will the .m file is run.
%The user will continue with the watermarking technique with the
%subsequent linkef menus
clc clear all

%Popup menu for the user to select choice
%Loop will always be true until the Exit is selected
while (1)
    input=menu('LSB Watermarking Main Menu','Start','Exit');
    switch input
        %Start
        case 1
            main1
            break
        %Close all the figures and exit from the popup menu
        case 2
            clear all
            close all
            break
    end
end
```

## C.2 Main1.m

```
%This program will allow the user to select the image to be watermark
%from the images available.
%The selected image will then be watermark with the watermarking
%technique selected in the next menu.
%Popup menu for the user to select choice
while (1)
    input1=menu('Select Image','Bird','Lena','Clock','Bridge','Camera',...
    'Return to Main Menu');
    switch input1
        %Select the bird image to be watermarked and proceed to main2.m
        case 1
            original_image='bird.bmp';
            main2
            break
        %Select the lena image to be watermarked and proceed to main2.m
        case 2
            original_image='lena.bmp';
            main2
            break
        %Select the clock image to be watermarked and proceed to main2.m
        case 3
            original_image='clock.bmp';
            main2
            break
        %Select the bridge image to be watermarked and proceed to main2.m
        case 4
            original_image='bridge.bmp';
            main2
            break
        %Select the camera image to be watermarked and proceed to main2.m
        case 5
            original_image='camera.bmp';
            main2
            break
        %Return to main
        case 6
            close all
            main
            break;
    end
end
```

## C.3 Main2.m

```
%This program allow the use to embed either a single or multiple
%watermark into the image selected.
%The watermarked image is then send to the main3.m to allow user
%to select the compression quality
%Popup menu for the user to select choice
while (1)
    input2=menu('Select Watermarking Scheme','Single Watermark',...
        'Multiple Watermark','Return to Image Selection');
    switch input2
        %Embed the single watermark into the selected image
        case 1
            single_embed
            main3
            break
        %Embedded multiple watermark into the selected image
        case 2
            multiple_embed
            main3
            break
        %Return to main1
        case 3
            close all
            main1
            break
    end
end
```

## C.4 Main3.m

```

%This program allow the user to select the JPEG compression
%quality after the image is watermarked
%using either the single or multiple watermarks.
%The compressed image will be passed to the decoder to decode
%the watermark

%Popup menu for the user to select choice
while (1)
    input3=menu('Select JPEG Compression Quality','100','99',...
        '98','95','90','Return to watermarking scheme');
    switch input3
        %To select quality factor of 100
        case 1
            imwrite(watermarked_image,cmap,'lsb_image.jpg','Quality',100)
            if input2==1
                main4
                break
            else
                main5
                break
            end
        %To select quality factor of 99
        case 2
            imwrite(watermarked_image,cmap,'lsb_image.jpg','Quality',99)
            if input2==1
                main4
                break
            else
                main5
                break
            end
        %To select quality factor of 98
        case 3
            imwrite(watermarked_image,cmap,'lsb_image.jpg','Quality',98)
            if input2==1
                main4
                break
            else
                main5
                break
            end
        %To select quality factor of 95
        case 4
            imwrite(watermarked_image,cmap,'lsb_image.jpg','Quality',95)
            if input2==1
                main4
                break
            else
                main5
                break
            end
        %To select quality factor of 90
        case 5
            imwrite(watermarked_image,cmap,'lsb_image.jpg','Quality',90)
            if input2==1
                main4

```

---

```
        break
    else
        main5
        break
    end
    %Return to main2
    case 6
        close all
        main2
        break;
    end
end
```

## C.5 Main4.m

```
%This program allow the user to decode the watermark using the
%single watermark decode as single
%watermark embedding was selected earlier in main2.m
%Popup menu for the user to select choice
while (1)
    input4=menu('Single Embed Watermark Retrieval','Decode',...
        'Return to Main Menu');
    switch input4
        %To decode the watermark
        case 1
            single_decode
        %Return to main
        case 2
            close all
            main
            break;
    end
end
```

## C.6 Main5.m

```
%This program allow the user to decode the watermark with or
%without a comparator watermark decode as multiple
%watermark embedding was selected earlier in main2.m
%Popup menu for the user to select choice
while (1)
    input5=menu('Multiple Embed Watermark Retrieval','Decode with ...
        Comparator','Decode without Comparator','Return to Main Menu');
    switch input5
        %Decode the watermark with the use of a comparator
        case 1
            with_comparator
        %Decode the watermark without the use of a comparator
        case 2
            without_comparator
        %Return to main
        case 3
            close all
            main
            break;
    end
end
```



## C.7 Single\_embed.m

```
%This program embed a single watermark into the image selected.
clc c=1; d=1; e=1;
%To read in the test image
[matrix_image cmap]=imread(original_image);
matrix_image=double(matrix_image); subplot(2,2,1)
image(matrix_image) colormap(cmap) title('Original Image')
%To read in the copyright image
[copyright dmap]=imread('copyright_small.bmp'); subplot(2,2,3)
image(copyright) colormap(dmap) title('Watermark')
%Convert to double for normalization, then back again
copyright=double(copyright); copyright=round(copyright./256);
copyright=uint8(copyright);
%To measure the size of the image
matrix_image_size=size(matrix_image); x=matrix_image_size(1);
y=matrix_image_size(2);
%To convert the matrix image from a MxN matrix into a row
for a=1:matrix_image_size(1,1)
    image_row(1,c:x)=matrix_image(a,1:y);
    c=c+y;
    x=x+y;
end
%To measure the size of the copyright image
copyright_size=size(copyright); o=copyright_size(1);
p=copyright_size(2); q=copyright_size(2);
%To convert the copyright image from a MxN matrix into a row
for b=1:copyright_size(1,1)
    copyright_row(1,d:q)=copyright(b,1:p);
    d=d+p;
    q=q+p;
end copyright_row_size=size(copyright_row);
%To generate the secret key of random numbers based on the
%copyright size
M=round(rand(copyright_size(1),copyright_size(2))*matrix_image_size(1)
*matrix_image_size(2));
M_size=size(M); o=M_size(1); p=M_size(2); q=M_size(2);
%To convert the M random numbers from a MxN matrix into a row
for b=1:M_size(1,1)
    M_row(1,e:q)=M(b,1:p);
    e=e+p;
    q=q+p;
end imagerow=image_row;
%To embed one bit of the watermark into the LSB bit of the
%chosen pixel of the image
for a=1:copyright_row_size(2)
    value=M_row(1,a);
    image_pixel=imagerow(value);
    copyright_pixel=copyright_row(1,a);
    imagerow(value)=bitxor(image_pixel,copyright_pixel);
end
```

---

```
i=1; j=y;
%To convert the watermarked image from a row vector back
%into a NxM matrix
for a=1:y
    watermarked_image(a,1:y)=imagerow(1,i:j);
    i=i+y;
    j=j+y;
end
end watermarked_image=uint8(watermarked_image);
subplot(2,2,2)
image(watermarked_image)
colormap(cmap)
title('Watermarked image')
```

## C.8 Multiple\_embed.m

```
%This program embed a multiple watermark into the image selected.
clc c=1; d=1; e=1;
%To read in the test image
[matrix_image cmap]=imread(original_image);
matrix_image=double(matrix_image); subplot(2,2,1)
image(matrix_image) colormap(cmap) title('Original Image')
%To read in the copyright image
[copyright dmap]=imread('copyright_small.bmp'); subplot(2,2,3)
image(copyright) colormap(dmap) title('Watermark')
%Convert to double for normalization, then back again
copyright=double(copyright); copyright=round(copyright./256);
copyright=uint8(copyright);
%To measure the size of the image
matrix_image_size=size(matrix_image); x=matrix_image_size(1);
y=matrix_image_size(2); x_block8=x/8; x_length8=x_block8*y;
count=0; counter=1;
%To convert the matrix image from a MxN matrix into 8 smaller
%MxN matrix
for b=1:8
    for a=1:matrix_image_size(1,1)
        image_row(b,c:x)=matrix_image(counter,1:y);
        c=c+y;
        x=x+y;
        count=count+1;
        counter=counter+1;
        if count==x_block8
            break
        end
    end
    c=1;
    x=y;
    count=0;
end
%To measure the size of the copyright image
copyright_size=size(copyright); o=copyright_size(1);
p=copyright_size(2); q=copyright_size(2);
%To convert the copyright image from a MxN matrix into a row
for b=1:copyright_size(1,1)
    copyright_row(1,d:q)=copyright(b,1:p);
    d=d+p;
    q=q+p;
end copyright_row_size=size(copyright_row);
%To generate the secret key of random numbers based on
%the copyright size
for a=1:8
    M=randperm(copyright_size(1)*copyright_size(2));
    M_row(a,:)=round((M*matrix_image_size(1)*x_block8)/
        (copyright_size(1)*copyright_size(2)));
end imagerow=image_row;
%To embed one bit of the watermark into the LSB bit of the
%chosen pixel of the image
for b=1:8
```

```
    for a=1:copyright_row_size(2)
        if M_row(b,a)==0
            M_row(b,a)=1;
        end
        value=M_row(b,a);
        image_pixel=imagerow(b,value);
        copyright_pixel=copyright_row(1,a);
        imagerow(b,value)=bitxor(image_pixel,copyright_pixel);
    end
end
i=1; j=y; count=1;
%To convert the watermarked image back to a MxN matrix
for a=1:8
    for b=1:x_block8
        watermarked_image(count,1:y)=imagerow(a,i:j);
        i=i+y;
        j=j+y;
        count=count+1;
    end
    i=1;
    j=y;
end watermarked_image=uint8(watermarked_image);
subplot(2,2,2)
image(watermarked_image)
colormap(cmap)
title('Watermarked image')
```

## C.9 Single\_decode.m

```
%This program decode the watermark from the single
%embed watermarked image

clc;
c=1;
t=1;

%To read in the watermarked image
[watermarked_image cmap]=imread('lsb_image.bmp');
watermarked_image=double(watermarked_image);

%To measure the size of the watermarked_image
watermarked_image_size=size(watermarked_image);
x=watermarked_image_size(1); z=watermarked_image_size(1);
y=watermarked_image_size(2);

%To convert the watermarked image from a MxN matrix into a row
for a=1:watermarked_image_size(1,1)
    watermarked_image_row(1,c:z)=watermarked_image(a,1:y);
    c=c+y;
    z=z+y;
end

%To retrieve the copyright image
for a=1:copyright_row_size(2)
    orig_pixel_value=image_row(M_row(1,a));
    watermarked_pixel_value=watermarked_image_row(M_row(1,a));
    retrieve_bits(1,a)=bitxor(orig_pixel_value,...
    watermarked_pixel_value);
end retrieve_bits=round(retrieve_bits.*256);

i=1; j=copyright_size(2);

%To convert the retrieve copyright image bits back to
%an MxN matrix
for a=1:copyright_size(1)
    retrieve(a,1:copyright_size(2))=retrieve_bits(1,i:j);
    i=i+copyright_size(2);
    j=j+copyright_size(2);
end

subplot(2,2,4) image(retrieve) colormap(dmap) title('Watermark
Decoded') retrieve=uint8(retrieve);
imwrite(retrieve,dmap,'watermark_image.bmp')
```

## C.10 With\_comparator.m

```

%This program decode the watermark using a comparator from
%the multiple embed watermarked image
clc c=1; t=1;
%To read in the watermarked image
[watermarked_image cmap]=imread('lsb_image.jpg');
watermarked_image=double(watermarked_image);
%To measure the size of the watermarked_image
watermarked_image_size=size(watermarked_image);
x=watermarked_image_size(1); y=watermarked_image_size(2);
count=0; counter=1;
%To convert the matrix image from a MxN matrix into 8
%smaller MxN matrix
for b=1:8
    for a=1:watermarked_image_size(2)
        watermarkedimage(b,c:x)=watermarked_image(counter,1:y);
        c=c+y;
        x=x+y;
        count=count+1;
        counter=counter+1;
        if count==x_block8
            break
        end
    end
    c=1;
    x=y;
    count=0;
end
%To retrieve the copyright image
for b=1:8
    for a=1:copyright_row_size(2)
        orig_pixel_value=image_row(b,M_row(b,a));
        watermarked_pixel_value=watermarkedimage(b,M_row(b,a));
        retrieve_bits(b,a)=bitxor(orig_pixel_value,...
            watermarked_pixel_value);
    end
end
%Comparator to compare all the watermark retrieve and determine
%the final watermark decoded
count_1=0; count_0=0; for b=1:108
    for a=1:8
        data=retrieve_bits(a,b);
        if data > 0
            count_1=count_1+1;
        else
            count_0=count_0+1;
        end
    end
    if count_1>count_0
        counter(1,b)=1;
    else
        counter(1,b)=0;
    end
    count_1=0;
    count_0=0;
end
retrieve_bits=counter; retrieve_bits=round(retrieve_bits.*256);

```

---

```
i=1; j=copyright_size(2);
%To convert the retrieve copyright image bits back to an MxN matrix
for a=1:copyright_size(1)
    retrieve(a,1:copyright_size(2))=retrieve_bits(1,i:j);
    i=i+copyright_size(2);
    j=j+copyright_size(2);
end
subplot(2,2,4) image(retrieve) colormap(dmap) title('Watermark
Decoded')
```

## C.11 Without\_comparator.m

```
%This program decode the watermark without using a comparator
%from the multiple embed watermarked image

clc;
c=1;
t=1;

%To read in the watermarked image
[watermarked_image cmap]=imread('lsb_image.jpg');
watermarked_image=double(watermarked_image);

%To measure the size of the watermarked_image
watermarked_image_size=size(watermarked_image);
x=watermarked_image_size(1); y=watermarked_image_size(2);
count=0; counter=1;

%To convert the matrix image from a MxN matrix into 8
%smaller MxN matrix
for b=1:8
    for a=1:watermarked_image_size(2)
        watermarkedimage(b,c:x)=watermarked_image(counter,1:y);
        c=c+y;
        x=x+y;
        count=count+1;
        counter=counter+1;
        if count==x_block8
            break
        end
    end
    c=1;
    x=y;
    count=0;
end

%To retrieve the copyright image
for b=1:8
    for a=1:copyright_row_size(2)
        orig_pixel_value=image_row(b,M_row(b,a));
        watermarked_pixel_value=watermarkedimage(b,M_row(b,a));
        retrieve_bits(b,a)=bitxor(orig_pixel_value,...
            watermarked_pixel_value);
    end
end

%Decode the watermark retrieve without a comparator
retrieve_bits=sum(retrieve_bits);
retrieve_bits=round(retrieve_bits./8);
retrieve_bits=round(retrieve_bits.*256);

i=1; j=copyright_size(2);

%To convert the retrieve copyright image bits back to
%an MxN matrix
for a=1:copyright_size(1)
    retrieve(a,1:copyright_size(2))=retrieve_bits(1,i:j);
    i=i+copyright_size(2);
    j=j+copyright_size(2);
end

subplot(2,2,4)
image(retrieve)
colormap(dmap)
```



---

```
title('Watermark  
Decoded')
```