Low-resolution watermarking for print security

This item was submitted to Loughborough University's Institutional Repository by the author.

Additional Information:


Metadata Record: https://dspace.lboro.ac.uk/2134/34004

Publisher: © Khaled Mahmoud

Rights: This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) licence. Full details of this licence are available at: https://creativecommons.org/licenses/by-nc-nd/4.0/

Please cite the published version.
University Library

Author/Filing Title ...........................................

Class Mark ............................................................

Please note that fines are charged on ALL overdue items.
Low Resolution Watermarking for Print Security

by

Khaled Walid Mahmoud

Doctoral Thesis
Submitted in partial fulfilment of the requirements
for the award of
Doctor of Philosophy of Loughborough University

December 2004

©by (Khaled Mahmoud) (2004)
Abstract

The problem of counterfeiting with scanners and printers has been substantially increased over recent years. This is largely due to the dramatic improvement in personal computer hardware and peripheral equipment. There has been a correspondingly increasing demand for digital watermarking techniques which can be applied to printed documents that prevent unauthorized copying of their content and at the same time, can withstand a substantial amount of abuse and degradation before and during scanning.

In this thesis, a new approach to digital watermarking a printed documents is presented. The process is defined by using analytical techniques and concepts borrowed from Cryptography. It is based on computing a 'scramble image' using convolution of the watermark image (logo) with a source of noise in a process termed 'Diffusion'. The cover image (text) is then inserted into the document's foreground using a simple additive process termed 'Confusion'. The watermark is subsequently recovered by removing the foreground, and then correlating with the original noise source.

It is demonstrated that this method is robust to a wide variety of attacks including geometric attacks, drawing, crumpling, and print/scan attacks. Experiments also show that the method is relatively insensitive to lossy compression, making it well suited to electronic document watermarking. The details of this method as well as the experimental results are shown.
The thesis is composed of seven chapters. Chapter 1 introduces digital watermarking; its general framework, principal applications, important properties, and the main aspects used to classify watermarking. Furthermore, some of the principal attacks that a watermarking system may face are discussed. This is followed by considering the human visual system in terms of its perception with regard to watermarking as well as some open problems. Chapter 2 provides a literature review of digital watermarking in frequency domain. First, the main features of the frequency domain that make it more appropriate for watermarking are explored. Next, three sub-domains are introduced (the Discrete Cosine Transform, the Discrete Wavelet Transform and the Discrete Fourier Transform). For each sub-domain, its properties as well as some techniques used to hide watermarks are discussed. Chapter 3 is concerned with how images are formed. This is because it is critical that we understand the physical principles of an imaging system and the theory of image formation in order to derive image encryption and watermarking model that are compatible with image capture devices (digital cameras and scanners for example) that, by default, conform to the 'physics' of optical image formation. Chapter 4 discussed the coding model for low-resolution watermarking developed for this thesis. This approach is extended and tested against the print/scan attacks presented in Chapter 5. Chapter 6 tests the system against some further practically important attacks. Finally, in Chapter 7, some applications that can benefit from this system are demonstrated. A short conclusion to the thesis and some ideas for future work and further development(s) are provided. Some further aspects of the work including the basis of optical diffraction theory and the software designed as part of this research are given in the Appendices. All the software developed is provided in a CD given at the back of this thesis.
Acknowledgements

My grateful thanks to God, for giving me the ability, health and knowledge to complete this work.

I am very grateful for the dedicated assistance provided by Professor J. M. Blackledge, Dr. Sekharjit Datta, and Dr. James A. Flint and for their constant help and guidance throughout the process of undertaking this research project.

I would also express my gratitude to Zarqa Private University (Jordan) for their financial support throughout the duration of my PhD studies and for providing a helpful and friendly working environment during my time in Jordan.

Finally, I wish to thank my parents, wife, and children, to whom I dedicate this thesis, for their encouragement, support, and all the sacrifices they made.
#Contents

Abstract i
Acknowledgements iii
Table of Contents iv
List of Figures x
List of Tables xiv
Notation xv

1 Digital Watermarking: Overview 1
  1.1 Introduction .......................... 2
  1.2 Watermarking Framework ............. 5
  1.3 Applications ........................... 6
  1.4 Classifications .......................... 8
     1.4.1 According to Inputs and Outputs .... 8
     1.4.2 According to Workspace Used ....... 9
     1.4.3 According to Visibility ............. 10
     1.4.4 According to Watermark Robustness .... 10
     1.4.5 According to Watermark Naturalness ... 11
1.5 Properties ........................................ 11
1.6 Distortions and Attacks .......................... 13
  1.6.1 Craver Classification for Attacks ......... 14
  1.6.2 Cox Classification for Attacks ............ 17
1.7 Related Disciplines and Subjects ............... 18
  1.7.1 Watermarking against other Techniques ... 19
  1.7.2 Problems with Cryptography as a Solution .. 19
  1.7.3 The Human Visual System (HVS) .......... 20
1.8 Some Open Problems ............................. 23

2 Survey of Watermarking in the Frequency Domain 25
  2.1 Transform Domain General Features .......... 26
  2.2 Human Visual System and Frequency Domain .. 27
  2.3 Frequency Domain Transforms ............... 28
    2.3.1 Discrete Cosine Transform ............. 29
    2.3.2 Discrete Wavelet Transform .......... 34
    2.3.3 Discrete Fourier Transform .......... 40
  2.4 Discussion .................................... 46

3 Optical Image Formation .......................... 47
  3.1 Digital Image Capturing ....................... 48
  3.2 Image Formation Systems ..................... 49
  3.3 Fourier Optics ................................ 52
    3.3.1 Linear System Properties .............. 52
    3.3.2 Optical Systems as Linear Systems .... 54
  3.4 Illumination and Image Formation ............ 58
    3.4.1 Incoherent Image Formation ........... 58
    3.4.2 Coherent Image Formation .............. 61
<table>
<thead>
<tr>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2.2 High-Pass Filtering</td>
</tr>
<tr>
<td>6.3 Additive Noise</td>
</tr>
<tr>
<td>6.4 Amplitude Effects</td>
</tr>
<tr>
<td>6.5 Thresholding</td>
</tr>
<tr>
<td>6.6 Geometric Distortion</td>
</tr>
<tr>
<td>6.6.1 Scaling</td>
</tr>
<tr>
<td>6.6.2 Cropping</td>
</tr>
<tr>
<td>Redundant Effects of Convolution</td>
</tr>
<tr>
<td>6.6.3 Spatial Translation</td>
</tr>
<tr>
<td>6.6.4 Rotation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 Applications, Future Work and Conclusions</td>
</tr>
<tr>
<td>7.1 Applications</td>
</tr>
<tr>
<td>7.1.1 Authentication</td>
</tr>
<tr>
<td>7.1.2 Transaction Tracking</td>
</tr>
<tr>
<td>7.1.3 Owner Identification (Copyright)</td>
</tr>
<tr>
<td>7.1.4 Steganography and Cryptography</td>
</tr>
<tr>
<td>7.2 Future Work</td>
</tr>
<tr>
<td>7.2.1 Additive Security Barrier</td>
</tr>
<tr>
<td>7.2.2 A Further Security Barrier</td>
</tr>
<tr>
<td>7.3 Conclusions</td>
</tr>
</tbody>
</table>

Appendices | 197 |
| A Diffraction Theory | 197 |
| A.1 Scalar Diffraction Theory | 198 |
| A.2 Kirchhoff Diffraction Theory | 199 |
| A.3 Fraunhofer Diffraction | 202 |
| A.4 Fresnel Diffraction | 205 |
# List of Figures

1.1 Data hiding example. ........................................... 3  
1.2 The general framework of a watermarking system. ............. 5  

3.1 Optical imaging system as a 'Black Box'. ......................... 50  
3.2 Diffraction geometry. ........................................... 64  
3.3 Lens-based image formation system. .......................... 68  
3.4 Optical system for spatial filtering. .......................... 71  

4.1 Printed valuable document. .................................... 77  
4.2 Watermarking model. .......................................... 81  
4.3 Embedding process. ............................................. 93  
4.4 Diffusion operators. ............................................ 96  
4.5 Generating the diffusion operator interface ....................... 97  
4.6 Generating diffused watermark interface. ...................... 98  
4.7 Extraction process. ............................................. 100  
4.8 Correlation interface. .......................................... 105  
4.9 Fresnel rings. .................................................. 106  
4.10 Grey-scale fresnel coding. .................................... 107  
4.11 Document fresnel coding. ..................................... 109  
4.12 Adaptive filtering (option 1) ................................ 110  
4.13 Adaptive filtering (option 2). ............................... 112
4.14 Adaptive filtering (options 4 and 5) .... 113

5.1 Original document used for Gaussian coding. .... 116
5.2 Watermark used in Gaussian coding. .... 117
5.3 Gaussian noise. .... 118
5.4 Gaussian coding: option 3 (without pre-processing) .... 120
5.5 Gaussian coding: option 1 .... 121
5.6 Gaussian coding: option 2 .... 122
5.7 Histogram analysis for 'removing foreground' algorithm .... 124
5.8 'Removing Foreground' interface .... 125
5.9 Damaged document .... 130
5.10 Extracted watermark from damaged document .... 131
5.11 Extracted watermark using Gaussian noise (seed=35) .... 131
5.12 Input images for noise coding .... 132
5.13 Uniform noise .... 133
5.14 Uniform coding: without pre-processing .... 134
5.15 Gamma noise .... 135
5.16 Gamma coding: without pre-processing .... 136
5.17 Poisson noise .... 137
5.18 Poisson coding: without pre-processing .... 138
5.19 Rayleigh noise .... 139
5.20 Rayleigh coding: without pre-processing .... 140
5.21 Fractal noise .... 142
5.22 Fractal coding: without pre-processing .... 143
5.23 Speckle noise .... 144
5.24 Speckle coding: without pre-processing .... 146
5.25 Original chaotic noise .... 147
5.26 Uniform chaotic noise .... 148
5.27 Chaos coding: without pre-processing .............................................. 149
5.28 Coloured text coding ................................................................. 151
5.29 Extracted watermark from coloured text document ......................... 152
5.30 Coloured background coding ....................................................... 152
5.31 Extracted watermark from coloured background document ............... 152

6.1 Attack experimental parameters ..................................................... 158
6.2 JPEG test results ......................................................................... 160
6.3 GLPF test results ......................................................................... 162
6.4 IHPF test results ........................................................................... 163
6.5 Additive noise test results ............................................................. 164
6.6 Amplitude change test results ....................................................... 165
6.7 Thresholding test results ............................................................... 167
6.8 Thresholding interface ................................................................. 168
6.9 Scaling test results ........................................................................ 169
6.10 Cropping test results-1 ............................................................... 170
6.11 Cropping test results-2 ............................................................... 170
6.12 Redundant effect of convolution-1 ................................................ 172
6.13 Redundant effect of convolution-2 ................................................ 173
6.14 Extracted watermark after translation .......................................... 174
6.15 Translation test results without cropping ....................................... 174
6.16 Rotation test results ................................................................. 175
6.17 Geometric distortion interface ..................................................... 177

7.1 ID card with diffused photo ........................................................... 179
7.2 Watermark: combination of photo and text .................................... 180
7.3 Coded statistical information ....................................................... 181
7.4 Coded specific part from a document ............................................. 182
7.5 Revealed document after coding specific parts ...
7.6 Gift voucher that have been tampered with ...
7.7 A watermarked exam ...
7.8 Extracted watermark from a coded exam ...
7.9 Watermarked cheque ...
7.10 A 2D barcode ...
7.11 Generating a 2D barcode texture field ...
7.12 Document header consisting of a logo and a semi-2D barcode ...
7.13 Document header consisting of a logo and a real 2D barcode ...
7.14 Extracted watermark for XOR coding ...
7.15 Extracted watermark for convolution coding ...
A.1 Diffraction of a plane wave ...
B.1 Main window interface ...
B.2 Main menu ...
B.3 RGB noise interface ...
B.4 ‘Select Two Images’ interface ...
B.5 ‘Add Two Images’ interface ...
B.6 ‘Metrics’ interface ...
B.7 Context menu ...
B.8 Padding techniques ...
B.9 Bar coding: 2D replacement (XOR) ...
B.10 2D barcode decoding (XOR) ...
B.11 ‘About’ interface ...
List of Tables

3.1 Principal properties of coherent and incoherent image formation 63

5.1 Extracted watermarks after Gaussian coding: electronic-version 127
5.2 Extracted watermarks after Gaussian coding: scanned-version. 129
5.3 Extracted watermarks after uniform coding. . . . . . . . . . . . 134
5.4 Extracted watermarks after Gamma coding. . . . . . . . . . . . 136
5.5 Extracted watermarks after Poisson coding . . . . . . . . . . . . . 138
5.6 Extracted watermarks after Rayleigh coding. . . . . . . . . . . . 140
5.7 Extracted watermarks after fractal coding . . . . . . . . . . . . . 143
5.8 Extracted watermarks after speckle coding . . . . . . . . . . . . . 146
5.9 Extracted watermarks after chaos coding . . . . . . . . . . . . . 149
## Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
<td>Wavelength of light in air.</td>
</tr>
<tr>
<td>$k$</td>
<td>Wavenumber.</td>
</tr>
<tr>
<td>$\omega$</td>
<td>Angular frequency.</td>
</tr>
<tr>
<td>⊗⊗</td>
<td>2D spatial convolution.</td>
</tr>
<tr>
<td>⊗⊗</td>
<td>2D spatial correlation.</td>
</tr>
<tr>
<td>$\oplus$</td>
<td>Exclusive-or operation</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>Standard deviation.</td>
</tr>
<tr>
<td>$\sigma^2$</td>
<td>Variance.</td>
</tr>
<tr>
<td>$\phi$</td>
<td>Phase component.</td>
</tr>
<tr>
<td>$w$</td>
<td>A watermark</td>
</tr>
<tr>
<td>$w'$</td>
<td>Extracted watermark.</td>
</tr>
<tr>
<td>$c$</td>
<td>Cover or foreground</td>
</tr>
<tr>
<td>$w_c$</td>
<td>Watermarked cover.</td>
</tr>
<tr>
<td>$\hat{w}$</td>
<td>Diffused watermark.</td>
</tr>
<tr>
<td>$\theta$</td>
<td>Theta angle.</td>
</tr>
<tr>
<td>$\Re[\cdot]$</td>
<td>Real part of a complex number.</td>
</tr>
<tr>
<td>$\mathcal{F}[\cdot]$</td>
<td>Fourier transform operator.</td>
</tr>
<tr>
<td>$|\cdot|$</td>
<td>Vector norm.</td>
</tr>
<tr>
<td>$\langle \cdot \rangle$</td>
<td>expected or mean value</td>
</tr>
<tr>
<td>Term</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>A/D</td>
<td>Analog to Digital.</td>
</tr>
<tr>
<td>COTF</td>
<td>Coherent Optical Transfer Function.</td>
</tr>
<tr>
<td>D/A</td>
<td>Digital to Analog.</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Fourier Transform</td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete Cosine Transform.</td>
</tr>
<tr>
<td>DWT</td>
<td>Discrete Wavelet Transform.</td>
</tr>
<tr>
<td>DVD</td>
<td>Digital Versatile Disk.</td>
</tr>
<tr>
<td>EZW</td>
<td>Embedded Zero Tree Wavelet.</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform.</td>
</tr>
<tr>
<td>GLPF</td>
<td>Gaussian Low Pass Filter.</td>
</tr>
<tr>
<td>HAS</td>
<td>Human Auditory System.</td>
</tr>
<tr>
<td>HVS</td>
<td>Human Visual System.</td>
</tr>
<tr>
<td>IDFT</td>
<td>Inverse Discrete Fourier Transform</td>
</tr>
<tr>
<td>IFFT</td>
<td>Inverse Fast Fourier Transform.</td>
</tr>
<tr>
<td>IHPF</td>
<td>Ideal High Pass Filter.</td>
</tr>
<tr>
<td>IOTF</td>
<td>Incoherent Optical Transfer Function</td>
</tr>
<tr>
<td>IRF</td>
<td>Impulse Response Function.</td>
</tr>
<tr>
<td>LSI</td>
<td>Linear and Shift Invariant System.</td>
</tr>
<tr>
<td>JND</td>
<td>Just Noticeable Distortion.</td>
</tr>
<tr>
<td>JPEG</td>
<td>Joint Photographic Experts Group.</td>
</tr>
<tr>
<td>MTF</td>
<td>Modulation Transfer Function.</td>
</tr>
<tr>
<td>OTF</td>
<td>Optical Transfer Function.</td>
</tr>
<tr>
<td>PSF</td>
<td>Point Spread Function.</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability Distribution Function.</td>
</tr>
<tr>
<td>Term</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>PTF</td>
<td>Phase Transfer Function.</td>
</tr>
<tr>
<td>RST</td>
<td>Rotation, Scaling, and Translation invariance</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-Noise Ratio.</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator.</td>
</tr>
<tr>
<td>XOR</td>
<td>Exclusive-or Operation.</td>
</tr>
</tbody>
</table>
Chapter 1

Digital Watermarking:
Overview

With rapid growth in computer networks and information technology, a large number of copyright works now reside in digital form. Furthermore, electronic publishing is becoming increasingly popular. These developments in computer technology increase the problems associated with copyright protection and enforcement and thus, future developments in networked multimedia systems are conditioned by the development of efficient methods to protect ownership rights against unauthorized copying and redistribution. Digital watermarking has recently emerged as a candidate to solve this difficult problem.

The goal of this chapter is to familiarize the reader with the main aspects of digital watermarking: framework, applications, properties, classifications, attacks, and etc.
1.1 Introduction

The mid-1990s saw the convergence of a number of different information protection technologies, whose theme was the hiding (as opposed to encryption) of information. Hiding can refer to either making the information imperceptible or keeping the existence of the information secret [1]. Figure 1.1 shows an example of data hiding. The original image (shown in figure 1.1 (a)) is protected by hiding a secret data (football image, figure 1.1 (c)) inside it without degrading the original image quality. The protected image is shown in Figure 1.1 (b). This hidden data can be extracted whenever it is necessary to prove the ownership of the image. Figure 1.1 (d) shows the extracted hidden data from the protected image.

Important sub-disciplines of information hiding are Steganography and Watermarking. Steganography and watermarking are concerned with techniques that are used to imperceptibly convey information. However, they are two different and distinct disciplines.

Watermarking is the practice of hiding a message (copyright notice or a serial number, for example) about an image, audio clip, video clip, or other work of media within that work itself [1] without degrading its quality in such a way that it is permanently embedded into the data and can be detected later. Steganography, on the other hand, is the study of the techniques used to hide one message inside another, without disclosing the existence of the hidden message or making it apparent to an observer that this message contains a hidden message [2]. From the previous definitions, they are distinguished as follows [1, 3]:

1. The information hidden by a watermarking system is always associ-
1.1 Introduction

Figure 1.1: Data hiding example. (a) The original image, (b) The protected image, (c) The hidden data, (d) The extracted hidden data

2 As the purpose of steganography is to have a covert communication between two parties whose existence is unknown to a possible attacker, a successful attack consists of detecting the existence of this communication. Watermarking, as opposed to steganography, has the additional requirement of robustness against possible attacks; even if the existence of the hidden information is known it should be hard for an attacker to destroy the embedded watermark. In other words, steganography is mainly concerned with detection of the hidden message while water-
1.2 Watermarking Framework

marking concerns potential removal by a pirate.

3 Steganographic communications are usually point-to-point (between sender and receiver) while watermarking techniques are usually one-to-many.

Nowadays, many websites use watermarking methods to protect their content (images, texts, videos, etc.). Moreover, they advertise this fact as a deterrent to copiers. For example, the following paragraph is copied from CORAL REEF CREATURES website (http://www.reefimages.com/) showing the copyright notice:

All photographs and text appearing in the Reef Images web site are the exclusive property of Doug Segar and Elane Stamman Segar. All photographs in the Reef Images web site are Watermarked with the http://www.digimarc.com Digimarc artist marking system. The photographs may not be reproduced, copied, stored, manipulated, republished electronically or in print without the written permission of Doug Segar or Elaine Stamman Segar.

The rest of this chapter is organized as follows. Section 1.2 gives the general framework of a digital watermarking system. Section 1.3 discusses several instances in which digital watermarking is already being used. Section 1.4 illustrates different aspects used in watermarking classification. Section 1.5 lists some important properties for watermarks. Section 1.6 describes how a digital watermarking system can be attacked. Section 1.7 introduces some related disciplines and subjects and finally Section 1.8 lists some open problems in this field.
1.2 Watermarking Framework

All watermarking schemes share the same generic building blocks (see Figure 1.2). These blocks and their functions are described below [4, 5].

**Watermark Embedding System (Signature Casting):** The embedded data is the watermark that one wishes to embed. It is usually hidden in a message referred to as a cover (work), producing the watermarked cover. The inputs to the embedding system are the watermark, the cover and an optional key. A key is used to control the embedding process so as to restrict detection and/or recovery of the embedded data to parties who know of it. The watermarked cover may face some intentional and/or unintentional distortion that may affect the existence of the watermark. The resultant outputs are called the 'Possibly Distorted Watermarked Cover'.

**Watermark Detection System (Extraction):** The inputs to the
1.3 Applications

detection system are the possibly distorted watermarked cover, the key and depending on the method, the original cover or the original watermark. Its output is either the recovered watermark or some kind of confidence measure indicating how likely it is for a given watermark at the input to be present in the work under inspection (e.g. Correlation).

Current watermarking schemes may be viewed as spread-spectrum communications systems [1], whose aim is to send the watermark between two parties with two sources of noise; noise due to the original cover and noise due to processing.

1.3 Applications

This section discusses some of the scenarios where watermarking is already being used as well as other potential applications. The list given here [1, 3, 4, 6] is by no means complete and is intended to give a perspective of the broad range of possibilities that digital watermarking opens.

**Owner identification.** Embedding the identity of a work's copyright holder as a watermark in order to prevent other parties from claiming the copyright of the data.

**Labelling.** The hidden message can contain labels that, for example, allow for annotation of images or audio data. Of course, the annotation may also be included in a separate file, but with watermarking it becomes more difficult to destroy or lose this label, since it becomes closely tied to the object that it annotates. This is especially useful in medical applications since it prevents potentially dangerous errors.
Fingerprinting (Transaction Tracking). This is similar to the previous application and allows acquisition devices (such as video cameras, audio recorders, etc) to insert information about the specific device (e.g., an ID number and date of creation). This is especially useful for identifying people who obtain the work legally but illegally redistribute it. This can involve the embedding of a different watermark into each distributed copy.

Authentication. Embedding signature information in a work that can be later checked to verify if it has not been tampered with.

Copy and Playback Control. The message carried by the watermark may contain information regarding copy and display permissions. A secure module can be added in copy or playback equipment to automatically extract this permission information and block further processing if required. In order to be effective, this protection approach requires agreements between work providers and consumer electronics manufacturers to introduce compliant watermark detectors in their video players and recorders. This approach is being taken in Digital Versatile Disks (DVD) for example.

Broadcast monitoring. Identifying when and where works are broadcast by recognizing watermarks embedded in the data.

Additional information. The embedded watermark could be an n-bit index to a database of URLs stored on a known location on the Internet. This index is used to fetch a corresponding URL from the database. The URL is then used to display the related web pages.
1.4 Classifications

Watermarking systems can be classified according to several aspects; some of which are listed below.

1.4.1 According to Inputs and Outputs

Private Marking Systems (Informed Detector)

These systems require at least the original cover in the detection side. This means that only the copyright holder can detect the watermark. In a private system, we can identify where the distortions were, and invert them before applying the watermark detector (using the original cover to reverse the embedding process or using the original work as a 'hint' to find where the watermark could be in the distorted watermarked cover). These types of systems may also require a copy of the embedded watermark for detection and just yield a 'YES' or 'NO' response to the question: does the distorted marked object contain this watermark?

Private systems usually feature increased robustness (greater strength to the embedded bits) not only toward noise-like distortions, but also distortions in the data geometry since it allows the detection and inversion of geometrical distortion [6]. Unfortunately, for these techniques to be applied, the possibility to access the original work must be granted. This means that the set-up of a watermarking system becomes more complicated, and on the other hand, the owners of the original works are compelled to insecurely share their works with anyone who wants to check the existence of the watermark.
Semi-private Marking Systems

These systems use the original watermark only and check whether it exists in the cover or not.

Public Marking Systems (Blind Marking)

These systems remain the most challenging since they require neither the secret original nor the embedded watermark. Blind watermarking techniques are less robust and are therefore more suitable for applications requiring lower security than copyright application, such as authorized copy distribution in electronic commerce.

1.4.2 According to Workspace Used

Another classification criterion distinguishes schemes into spatial domain techniques and transform-domain techniques depending on whether the watermark is encoded by directly modifying pixels (such as simply flipping low-order bits of selected pixels) or by altering some frequency coefficients obtained by transforming the image into the frequency domain. Spatial domain techniques are simple to implement and often require a lower computational cost, although they can be less robust against tampering than methods which place the watermark in the transform domain. Watermarking schemes that operate in a transform space are increasingly common, as this can aid robustness against several attacks and distortions (transform domain methods hide messages in significant areas of the cover image, which makes them more robust to attacks). Moreover, while they are more robust to various kinds of signal processing, they remain imperceptible to the human sensory
1.4 Classifications

Most schemes operate directly on the components of some transform of the cover, like the discrete cosine transform, discrete wavelet transform or discrete Fourier transform.

1.4.3 According to Visibility

Copyright marks do not always need to be hidden, as some systems use visible digital watermarks [4], but most of the literature has focused on invisible (or transparent) digital watermarks which have wider applications. Modern visible watermarks may be visual patterns (e.g. a company logo or copyright sign) overlaid on digital images.

1.4.4 According to Watermark Robustness

Fragile Watermarks

Watermarks that have very limited robustness and are destroyed as soon as the object is modified too much. They are applied to detect modifications of the watermarked data, rather than conveying unreadable information [1]. Cryptographic techniques have already made their mark on authentication. However, there are two significant benefits that arise from using watermarking: First, the signature becomes embedded in the message. Second, it is possible to create ‘soft authentication’ algorithms that offer a multi-valued measure that accounts for different unintentional transformations that the data may have suffered instead of the classical yes/no answer given by cryptography-based authentication.
Robust Watermarks

Robust watermarks have the property that it is not feasible to remove them or make them useless without destroying the object at the same time. This usually means that the mark should be embedded in the most robust significant components of the object [6].

1.4.5 According to Watermark Naturalness

Watermarks that range from pseudo-random sequences to small image logo that can be easily recovered and authenticated

1.5 Properties

Watermark systems can be characterized by a number of defining properties [1, 6, 4]. The relative importance of each property is dependent on the requirements of the application and the role that the watermark will play. Some important properties are listed below:

Fidelity (watermark imperceptibility) Perceptual similarity between the original and the watermarked versions must be very high (i.e. the difference between the original work and the watermarked work should be invisible). It has been argued that the watermark should not be noticeable to the viewer instead of being imperceptible [6]. Furthermore, if a signal is truly imperceptible, then perceptually base lossy compression algorithms should, in principle, remove such a signal. Current compression algorithms can still leave room for an imperceptible signal to be inserted. This may not be true of the next generation of compression
algorithms. Thus, to survive the next generation of lossy compression algorithms, it is necessary for a watermark to be noticeable to a trained observer.

**Statistical invisibility** The watermark must be statistically invisible to thwart unauthorized removal (i.e. a statistical analysis should not produce any advantage from the attacking point of view) The noise-like watermark is statistically invisible and has good auto-correlation properties

**Readily extracted** If the decoder needs to run in real-time, then it is necessary for the decoding process to be significantly simpler than the encoding process [6]. In some applications, this requirement is reversed depending on the purpose of watermarking system.

**Data payload** This refers to the amount of information that can be carried in a watermarked cover and raises capacity issues in digital watermarking. The length of the watermark serves as a measure of the capacity. A longer watermark signal means that more coefficients need to be modified; hence, the watermarked images look 'noisier'. The more information one wants to embed, the lower the watermark robustness.

**Embedding Effectiveness** The probability that the embedder will successfully embed a watermark in a randomly selected work. This property is related to real-time embedding system (which must be high).

**False Positive Rate** The frequency with which we should expect a watermark to be detected in a non-watermarked object (which must be low).

**Robustness (Security)** The watermark should be resilient to standard manipulations which are both intentional and unintentional in nature.
1.6 Distortions and Attacks

Some authors [1] distinguish between resistance to intentional and unintentional attacks. They use 'security' when dealing with the ability of the watermark to resist hostile attacks while using the 'robustness' when dealing with the ability of the watermark to survive normal processing of the work such as spatial filtering, lossy compression, printing and scanning, geometric distortions (such as rotation, translation and scaling). Note, robustness actually comprises two separate issues: (i) whether or not the watermark is still present in the data after distortion; (ii) whether the watermark detector can detect it. For example, watermarks inserted by many algorithms remain in the data after geometric distortions but the corresponding detection algorithm can only detect the watermark if the distortion is first removed, otherwise the detector can not detect the watermark [6]. In general, any increase in robustness comes at the expense of increased watermark visibility. Also the presence of the original cover increases the robustness. For example, the use of the original work permits some pre-processing to be carried out before the watermark checking such as: rotation angles, translation and scaling factors which can usually be estimated, together with missing parts of the image which can be replaced by corresponding parts of the original one. It is possible to undertake an exhaustive search on different rotation angles and scaling factors until a watermark is found, but this is prohibitively computationally intensive.

1.6 Distortions and Attacks

In practice, a watermarked cover may be altered either intentionally or unintentionally, so the watermarking system should still be able to detect and
1.6 Distortions and Attacks

extract the watermark. The distortions are limited to those that do not produce excessive degradations, since otherwise the transformed object would be unusable. Several authors have classified attacks based on a range of aspects. One famous classification has been carried out by Craver et al. [7, 8].

1.6.1 Craver Classification for Attacks

Craver defines four general classes of attacks, organized by the way in which the attackers try to defeat the watermarking technology. These classes are illustrated below together with some examples for each class. Some of them may be intentional or unintentional, depending on the application.

Robustness Attacks (Unauthorized removal)

This type of attacks aim to diminish or remove the presence of a digital watermark from its associated work, while preserving the work so that it is not useless after the attack is over. Some examples of robustness attacks are discussed below.

- Additive Noise: This may happen (unintentionally) in certain applications such as D/A (printing) and A/D (scanning) converters or from transmission errors. It could happen intentionally by an attacker who is trying to destroy the watermark (or make it undetectable) by adding noise to the watermarked cover.

- Filtering: Linear filtering such as low-pass filtering or non-linear filtering such as median filtering.

- Collusion attack: In some watermarking schemes, if a work has been
1.6 Distortions and Attacks

watermarked many times under different secret keys, it is possible to collect many such copies and ‘average’ them into a composite work that closely resembles the original one and does not contain any useful watermarking data [9].

- Inversion attack (elimination attack): An attacker may try to estimate the watermark and then remove the watermark by subtracting the estimate or reverse the insertion process to perfectly remove the watermark. This means that an attacked object can not be considered to contain a watermark at all (even using a more sophisticated detector). Note, that with different watermarked objects, it is possible to improve the estimate of the watermark by simple averaging.

- Lossy Compression. This is generally an unintentional attack, which appears very often in multimedia applications. Practically, nearly all audio, video and digital images that are currently distributed via the Internet are in compressed form. Lossy image compression algorithms are designed to disregard redundant perceptually-insignificant information in the coding process. Watermarking tries to add invisible information to the image. An optimal image coder would therefore simply remove any embedded watermark information. However, even state-of-the-art image coding such as JPEG 2000 does not achieve optimal coding performance and therefore there is a ‘distortion gap’ that can be exploited for watermarking. Actually, one can observe that the use of a particular transform provides good results against compression algorithms based on the same transform. For instance, DCT-domain image watermarking is more robust to JPEG compression than spatial-domain watermarking.
Presentation Attacks (Masking Attacks)

This attack does not attempt to remove the watermark, but instead alters the work so that the watermark can no longer be detected or extracted easily. This means that the attacked work can still be considered to contain the watermark, but the watermark is undetectable by an existing detector (such as a detector sensitive to image rotation). Examples of presentation attacks include:

- **Chopping attack (mosaic attack)** Here, an image is 'chopped' into distinct sub-images, which are embedded one after another in a web page. Common web browsers render sub-images together as a single image, so the result is identical to the original image. However, the chopping process distributes the original image's watermark into many pieces and the watermark cannot be recovered unless the original image is reconstructed first.

- **Rotation and Spatial Scaling.** Detection and extraction fail when rotation or scaling is performed on the watermarked image because the embedded watermark and the locally generated version do not share the same spatial pattern anymore. This kind of attack can be unintentional, occurring during the scanning-printing process (copies from printing and/or scanning maybe rotated, scaled, cropped or translated in comparison with the original image).

- **Cropping.** This is a very common attack since in many cases the attacker is interested in a small portion of the watermarked object, such as parts of a certain picture or frames of a video sequence. With this in mind, in order to survive this kind of attack, the watermark needs
Interpretation Attacks

This kind of attack seeks to forge invalid or multiple interpretations from watermark evidence [7] whereby an attacker can devise a situation, which prevents assertion of ownership. Some example are:

- Multiple watermarking. An attacker may watermark an already watermarked object (creating uncertainty about which watermark was inserted first) and later make claims of ownership. The easiest solution is to time-stamp the hidden information by a certification authority.

- Unauthorized embedding (Forgery). Embedding an illegitimate watermark into works that should not contain them or using watermark inversion to remove the original watermark before inserting a new watermark.

Legal Attacks

In a legal attack, the attacker uses a legal precedent, the identity or reputation of the object owner, or some other non-technical information to establish doubt in court as to whether a watermark actually constitutes the proof that its owner claims.

1.6.2 Cox Classification for Attacks

Cox et al. [1] classify the attacks into two main categories: active and passive attacks.
1.7 Related Disciplines and Subjects

Active Attacks

Active attacks change the cover, such as:

- Unauthorized removal (robustness attack).
- Unauthorized embedding (forgery).

Passive Attacks

Passive attacks do not change the cover, such as: Unauthorized detection which can be in three levels according to severity.

1. the adversary detects and deciphers an embedded message,
2. the adversary detects the watermark and distinguishes one mark from another, but cannot decipher what the marks mean,
3. the adversary detects the watermark but without distinguishing the decipher.

There are situations in which the watermark has no hostile enemies and need not to be secure, e.g. when the watermark is used to provide enhanced functionality.

1.7 Related Disciplines and Subjects

This section discusses other techniques used for information hiding and why watermarking is more powerful. Also, the importance of the Human Visual System (HVS) to watermarking is given.
1.7 Related Disciplines and Subjects

1.7.1 Watermarking against other Techniques

Watermarking is distinguished from other techniques (such as placing the mark in the media header, encoding it in a visible bar, or speaking it out loud as an introduction to an audio clip) in three ways:

1. a watermark is imperceptible;

2. a watermark is inseparable from the work (once the digital image is printed on paper, all data in the header are left behind, furthermore, this data may not survive a change in the image format);

3. a watermark undergoes same transformations on the work, which can help in authentication and detection based on the kind of alteration that the work has undergone.

1.7.2 Problems with Cryptography as a Solution

Cryptography can be defined as the study of secret writing, i.e. concealing the contents of a secret message by transforming the original message into a form that cannot be easily interpreted by an observer. Thus, the mere discovery of encrypted data suggests that something is illicit, or at least secret, is occurring. Cryptographic techniques can hide a message from plain view during communication, and can also provide auxiliary information that effectively proves the messages. However, traditional cryptosystems suffer from one important drawback, which renders them useless for the purpose of enforcing copyright law [8]: they do not permanently associate cryptographic information with work. Thus, cryptography alone cannot make any guarantees about the redistribution or alteration of work after it has initially
passed through the cryptosystem (i.e. cryptography can not help the seller monitor how a legitimate customer handles the work after decryption). Watermarking can fulfil this need; it places information within the work where it is never removed during normal usage. Furthermore, steganography has a distinct advantage over cryptography; it allows for the communication of secret information without alerting an attacker to the presence of the secrets.

1.7.3 The Human Visual System (HVS)

It is currently widely accepted that robust/high fidelity watermarking techniques should largely exploit the characteristics of the HVS and human auditory system (HAS) for more effectively hiding watermarks. Perceptual masking techniques exploit the perceptual masking properties of the human auditory system and of the human vision system [10].

HVS and Fidelity

A good watermarking method has to adapt to the particular image being watermarked in order to exploit specific HVS characteristics and hence amplifying the watermark where the alterations are least likely to be noticed. Local image characteristics that can help determine the visibility of a watermark are listed below:

1. Fine against high texture area: it is usually true to say that human eyes are not sensitive to the small changes in texture but are very sensitive to small changes in the smooth areas of an image. Hence, it should be possible to incorporate more information into those parts of the image that contain more textures than smooth area. Related methods
1.7 Related Disciplines and Subjects

accomplish this by calculating a value of local contrast and mapping increasing contrast values to increasing watermark magnitudes [11]

2 Edges: edge information of an image is the most important factor for perception of the image. This can present a problem though, as directional edges separating two distinct objects in an image may be identified as high contrast areas. This results in the application of a higher strength watermark signal around the connected edges, which causes objectionable watermark ringing on connected edges. Methods have also been proposed that identify areas of true high contrast texture while protecting connected directional edges [11, 12] (regions that contain a sudden transition in luminance).

3. Brightness Sensitivity: when the mean value of the square of the noise is the same as the background, the noise tends to be most visible against mid-grey backgrounds (i.e., the mid-grey regions have lower noise-capacity compared to other regions [12]).

HVS and Robustness

The key to making the watermark robust and to preventing the watermark from being easily attacked is to embed the watermark in the perceptually significant regions of the image. These regions do not change much after several signal processing or compression operations. Moreover, if these regions lose their fidelity significantly, the reconstructed image could be perceptually different from the original one (i.e., visual fidelity is only preserved if the perceptually significant regions remain intact). Also, lossy image compression algorithms are designed to disregard redundant information. Information bits placed within textured areas of the image are therefore more vulnerable
to attack. The question, therefore, is how much extra watermark information can be added to the perceptually significant regions without any impact on the visual fidelity? There is a compromise to be reached between hiding a large number of information bits where they can least be seen, but where they can be attacked by image compression algorithms, or placing fewer bits on less textured but safer portions of the image.

Measure of Capacity

Every pixel value of an image can be altered only to a certain limit without making a perceptible difference to the image quality. This limit can be called as the 'Just Noticeable Distortion' or JND level [13, 11, 12]. For instance, smooth areas are assigned relatively low JND compared to strongly textured regions (i.e. strongly textured regions have a very high capacity for noise).

Watermark Shaping

There is an advantage to shaping the watermark spectrum based on the cover to match currently known human visual systems. Inserting a watermark that is a function of the cover leads to a non-linear embedding procedure. Such a procedure has the advantage that when the image energy in a particular area is small, the watermark energy is also reduced, thereby avoiding artifacts, and when the image energy is large, the watermark energy is increased, thereby improving the robustness of the procedure. Conversely, if simple linear addition to the watermark and image occurs, then the energy of the watermark must be very low in order to avoid worst case scenarios in which the image energy in a particular place is very low and artifacts are created because the watermark energy was too strong relative to the image [6].
HVS and Spread Spectrum Techniques

Spread-spectrum techniques spread a narrow band signal (watermark) over a much wider band (cover) such that the signal-to-noise ratio in any single band is very low. However, with precise knowledge of the spreading function, the receiver is able to extract the transmitted signal, summing up the signals in each of the bands such that the detector signal-to-noise ratio is strong. Spread spectrum techniques are useful because they have a low probability of interception by an attacker [10]. Embedding a watermark in the high frequency spectrum yields low robustness, whereas embedding the watermark in the low frequency spectrum yields visible impacts. Spread spectrum techniques can reconcile these conflicting points by allowing a low-energy signal to be embedded in each of the frequency bands (both high and low).

1.8 Some Open Problems

Even though watermarking is a fast growing field there are a number of outstanding problems such as the following [14]:

- Optimization between robustness and visibility limiting the capacity.
- Detection speed, which is crucial especially in real time applications.
- Reading the watermark after geometric distortion.
- Adaptability to different printing processes, paper and ink which may degrade the watermark. Moreover, printed images do not maintain their quality over time. They are subject to aging, soiling, crumbling, tearing and deterioration. Designing a watermark scheme to compensate for these kinds of unintentional attacks is another challenge.
1.8 Some Open Problems

- Different input devices (scanner, cameras) introduce different types of distortions. Accounting for these differences in detection is also a major challenge.
Chapter 2

Survey of Watermarking in the Frequency Domain

An important classification distinguishes watermarking schemes into spatial domain techniques and transform domain techniques, depending on whether the watermark is encoded by directly modifying pixels (such as simply flipping low-order bits of selected pixels) or by altering some frequency coefficients obtained by transforming the image into the frequency domain. In Section 2.1, the main features of the frequency domain are explored. Subsequent sections introduce three frequency domains and their properties. Many watermarking methods in each domain are explained. These domains are the Discrete Cosine domain, the Discrete Wavelet domain, and Discrete Fourier domain. The Fourier domain is the basis for the methods developed and implemented as part of the work reported in this thesis.
2.1 Transform Domain General Features

Watermarking schemes that operate in a frequency space possess a number of desirable features since:

- By transforming spatial data into another domain, statistical independence between pixels as well as high-energy compaction can be obtained.

- The watermark is irregularly distributed over the entire spatial image upon an inverse transformation, which makes it more difficult for attackers to decode and read the mark.

- Markers can be provided according to the perceptual significance of different transform domain components, which means that the watermark can be adaptively placed where it is least noticeable, such as within a textured area. Moreover, transform domain methods can hide messages in significant areas of the cover which makes them more robust against many attacks and distortion. However, while they are more robust to various kinds of signal processing, they remain imperceptible to the human sensory system.

- Cropping may be a serious threat to any spatially based watermark but is less likely to affect a frequency-based scheme. Since watermarks applied to the frequency domain will be dispersed over the entirety of the spatial image upon inverse transformation. Therefore, it is possible to retrieve part of the watermark.

- Lossy compression is an operation that usually eliminates perceptually unimportant components of a signal. Most processing of this sort takes
place in the frequency domain. In fact, matching the transform with a compression transform may result in better performance of the data-hiding schema (i.e. DCT for JPEG, Wavelet for JPEG-2000)

- The characteristics of the Human Visual System (HVS) can be fully exploited in the frequency domain. This aspect will be explored further in the next section.

2.2 Human Visual System and Frequency Domain

It is usually the case that the human eye is not sensitive to small changes in edges and texture but is very sensitive to small changes in the smooth areas of an image. In flat featureless portions of the image, important information concerned with the 'flat parts' concentrate on the lowest frequency components, while, in a highly textured image, energy is concentrated in the high frequency components. Therefore, the human eye is more sensitive to lower frequency noise than high frequency noise. Taking into account these fundamental points:

- The watermark should be embedded into the higher frequency components to achieve better perceptual invisibility; however, high frequencies might be discarded after attacks such as lossy compression, shrinking or scanning.

- In order to prevent the watermark from being easily attacked, it is often necessary to embed it into the lower frequency coefficients. The attacker can not change these coefficients, otherwise the image can be
damaged. However, the human eye is more sensitive to lower frequency noise.

From the points discussed above, in order to invisibly embed a watermark, which can survive most attacks, a reasonable trade-off is possible by embedding the watermark into the middle frequency range of the image [15].

### 2.3 Frequency Domain Transforms

During watermarking in the transform domain, the original host data is transformed, and the transformed coefficients are perturbed by a small amount in one of several possible ways in order to represent the watermark. Coefficient selection is based on perceptual significance or energy significance. When the watermarked image is compressed or modified by any image processing operations, noise is added to the already perturbed coefficients. The private retrieval operation subtracts the received coefficients from the original ones to obtain the noise perturbation. The watermark is then estimated from the noisy data as best as possible. The most difficult problem associated with blind watermark detection in the frequency domain is to identify the coefficients used for watermarking. Embedding can be done by adding a pseudo-random noise field, quantization (threshold) or image (logo) fusion. Most algorithms consider HVS to minimize perceptibility. The aim is to place more information bits where they are most robust to attack and are least noticeable. Most schemes operate directly on the components of some transform of the cover such as the Discrete Cosine Transform (DCT), Discrete Wavelet Transforms (DWT) and Discrete Fourier Transforms (DFT). This section introduces each domain, illustrates its main features and introduces
some techniques that use the transformation in watermarking.

2.3.1 Discrete Cosine Transform

The DCT transform has a number of desirable properties in respect of watermarking:

- The DCT has the primary advantage that it is a sequence of real numbers, provided that the input sequence is real.

- The two-dimensional DCT is the basis for most popular lossy digital image compression system used today, e.g. the JPEG system.

- The sensitivity of HVS to the DCT basis images has been extensively studied, resulting in a default JPEG quantization table.

Embedding Techniques in the DCT Domain

Zhao and Koch [16] approach the problem by segmenting the image into $8 \times 8$ blocks. Block DCT transformation and quantization steps are applied on each block. A bit of information can be encoded in a block using the relation between three quantized DCT coefficients ($c1, c2, \text{ and } c3$) from this block. The three coefficients must correspond to the middle frequencies. One block encodes a '1', if $c1 > c3 + d$ and $c2 > c3 + d$. On the other hand, a '0' is encoded, if $c1 + d < c3$ and $c2 + d < c3$. The parameter $d$ accounts for the minimum distance between two coefficients. The higher $d$ is, the more robust the method will be against image processing techniques. If the relations between the coefficients do not correspond to the encoded bit, a change must be made to the coefficients so that they can represent
the encoded bit. If the modification required to code one bit of information is too large, then the block is not used and marked as an invalid block. Consequently, the blocks are de-quantized and the inverse DCT is applied. In the decoding step, comparing the three coefficients of every block in the quantized DCT domain can restore the label.

Cox, et al. [17] present an image watermarking method in which the mark (a sequence of real numbers \( W = \{w_i\} \) having a normal distribution with zero mean and a unity variance) is embedded in the \( n \) (excluding the DC term) most perceptually significant frequency components \( V = \{v_i\} \) of an image's DCT to provide greater robustness to JPEG compression. The watermark is inserted using the procedure \( \tilde{v}_i = v_i + \alpha * v_i * w_i \). This modulation law is designed to take into account the frequency masking characteristics of the human visual system. This non-linear insertion procedure adapts the watermark to the energy present in each coefficient. The advantage of this is that when \( v_i \) is small, the watermark energy is also small, thereby avoiding artifacts. When \( v_i \) is large, the watermark energy is increased for robustness. The parameter \( \alpha \) represents a compromise between robustness and image fidelity. The presence of the watermark is verified by extracting the main components of the original image and those with same index from a watermarked image and inverting the embedding formula to give a possibly modified watermark \( W' \). The watermark is said to be present if the correlation between \( W \) and \( W' \) is greater than a given threshold.

Barni, et al. [18] propose a watermarking algorithm similar to Cox's method. However, instead of using the \( n \) largest DCT coefficients as Cox does, the set is produced by arranging the DCT coefficients in a zigzag order and a subset in the mid-frequency range is selected. The lowest coefficients are then skipped to preserve perceptual invisibility. The watermark is then
2.3 Frequency Domain Transforms

embedded in this set of coefficients in the same way as Cox. In order to enhance the invisibility of the watermark, the spatial masking characteristics of the HVS are also exploited to adapt the watermark to the image being signed: the original image ($I$) and the watermarked image ($I'$) are added pixel by pixel according to a local weighting factor $b_{i,j}$, thus producing a new watermarked image $I'_{i,j} = I_{i,j}(1 - b_{i,j}) + b_{i,j} * I_{i,j}'$. In regions characterized by low-noise sensitivity, where the embedding of the watermarking data is easier (e.g. highly textured regions), $b_{i,j} \approx 1$, i.e. the watermark is not diminished. In regions more sensitive to change, in which the insertion of the watermark is more disturbing (e.g. uniform regions), $b_{i,j} \approx 0$, i.e. the watermark is embedded only to a minor extent. In the extraction phase, one first extracts the subset of modified coefficients from the full frame DCT of the watermarked image. The correlation between the marked (possibly corrupted coefficients) and the mark itself is taken as a measure of the mark presence.

O' Ruanaidh, et al. [19] present a private watermarking technique for images using bi-directional coding in the DCT domain. In bi-directional coding, the image is divided up into blocks. The DCT is computed for each block. The mean of each block is incremented to encode a '1' or decremented to encode a '0'. This may be accomplished by using simple thresholding techniques. A JPEG quantization table (visual masking) is used to weight the DCT coefficients in each block. The most significant components are then selected by comparing the square of the component magnitude to the total energy in the block. In the decoding step, the mean of each block in the original non-watermarked image is compared with the mean of the corresponding blocks in the tested copy to decode the stored bit.

Swanson, et al. [20] embed the watermark by computing the DCT
for each block in the cover. A perceptual mask is computed for each block. The resulting perceptual mask is then scaled and multiplied by the DCT of the pseudo-noise watermark. The schema uses a different sequence for each block. The watermark is then added to the corresponding block. The watermark can be detected by correlating the modified watermark with the original watermark and comparing the result to a threshold.

Chae and Manjunath [21] use a public technique to embed a signature (watermark) into images. The signature's DCT coefficients are quantized according to a signature quantization matrix. The resulting quantized coefficients are encoded using lattice-codes. The choice of the signature quantization matrix affects the quantity and the quality of the embedded data. The codes are inserted into the middle frequency DCT coefficients of the host image. This insertion is adaptive to the local texture content of the host image blocks and is controlled by the block texture factor. The texture factor is computed using a wavelet transform. The selected host coefficients are then replaced by the signature codes and combined with the original unaltered DCT coefficients to form a fused block of DCT coefficients. The fused coefficients are then inverse-transformed to give an embedded image.

Bors and Pitas [22] propose a watermarking algorithm based on imposing constraints in the DCT domain. The block sites for embedding the watermark are selected based on a Gaussian network classifier, then the DCT constraints are embedded in the selected blocks. Two distinct algorithms are considered here. The first algorithm embeds a linear constraint on selected DCT frequency coefficients (i.e. $Y = FQ$, where $F$ is the vector of the modified DCT coefficients, and $Q$ is the weighting vector provided by the watermark). In the second approach, circular regions are defined around certain DCT coefficients. For a selected block site, they evaluate the Euclidean
distance between its DCT coefficient vectors and that of the watermark. The chosen DCT coefficients are changed to the value of the closest watermark parameter vector. After modifying the DCT values, the image is reconstructed based on the inverse DCT transform. In the detection stage, a check is made on the DCT constraints after which the respective block is located. A given site is considered as being 'signed' when the probability of detecting the constrained DCT coefficients and the probability of detecting the location constraint is maximized. The original image is not required for watermark detection and simulations have showed this method to be resistant to JPEG compression and filtering.

Kankanhalli, et al. [12] propose a way of analysing the noise sensitivity of every pixel based on the local region content (texture, edges and luminance). If the distortion caused by the watermarking algorithm is at or below a threshold, then the degradation in the original image quality is imperceptible. The analysis is based on the DCT coefficients. The energy in the DCT coefficients can be used as a measure of roughness, and the count of large-magnitude fluctuations in a high-energy block can then be used to decide if the block has an edge or is highly textured. The work also analyses the contribution of luminance to noise sensitivity. This luminance analysis is done at the pixel level in the spatial domain. The authors use the previous mask to embed an invisible watermark in the spatial domain.

Tao and Dickenson [23] have an approach that is similar to that of [12]. Here, the block as a whole is given a sensitivity label that shapes the watermark based on texture and edges analysis. The embedding is then done in the DCT transform domain.
2.3 Frequency Domain Transforms

2.3.2 Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) is similar to a hierarchical sub-band system, where the sub-bands are logarithmically spaced in frequency and represent octave-band decomposition. The DWT can be implemented using digital filters and down-samplers [24]. The original image is split into four quadrant bands after decomposition. The four quadrants contain approximations to the sub-band (LL), horizontal detail sub-band (LH), vertical detail sub-band (HL) and a diagonal detail sub-band (HH). This process can be repeatedly applied on the approximation sub-band to generate the next coarser scale wavelet coefficients. The process continues until some final scale is reached. The wavelets transform has a number of advantages [25, 26] over other transform that can be exploited for watermarking including the following:

- It is well known that wavelet coding has been exploited in new compression standards such as JPEG2000 and MPEG4 due to the excellent performance in compression.

- The wavelet transform requires a lower computational cost of $O(n)$ compared with the Fourier or the Cosine transform which are of $O(n \log_2 n)$, where $n$ is the length of the signal.

- The wavelet transform processes data at different scales or resolutions, highlighting both large and small features. This allows watermarking to become adaptive as it depends on the local image characteristic at each resolution level.

- The wavelet functions provide good space-frequency localization and thus, they are suited for analyzing images where most of the informative
content is represented by components localized in space such as edges and borders.

- With the DWT, the edges and texture are usually exploited very well in the high frequency sub-band (HH, HL and LH). Therefore, adding a watermark via these large coefficients is difficult for the human eye to perceive.

- Wavelet functions have advantages over traditional Fourier methods in analyzing signals containing many discontinuities or sharp changes.

- The wavelet transform is flexible enough to adapt to a given set of images or particular type of application. The decomposition filters (such as Haar, Daubechies-4, 6 or bi-orthogonal filters) and the decomposition structure (wavelet packet, complex wavelet transform) can be chosen to reflect the characteristics of the image.

- Research into human perception [27] indicates that the retina of the eye splits an image into several frequency channels each spanning a bandwidth of approximately one octave. The signals in these channels are processed independently. Similarly, in a multi-resolution decomposition, the image is separated into bands of equal bandwidth on a logarithmic scale. It can therefore be expected that use of the discrete wavelet transform will allow the independent processing of the resulting components without significant perceptible interaction between them, and hence makes the process of imperceptible marking more effective. For this reason the wavelet decomposition is commonly used for the fusion of images.
Embedding Techniques in the DWT Domain

There have and continue to be many attempts to use the wavelet transform in watermarking. These include the following.

Xia, et al. [25] propose a private watermarking system. The method utilizes large DWT coefficients of all sub-bands excluding the approximation image to equally embed a random Gaussian distributed watermark sequence in the whole image. The decoding process is based on a hierarchical correlation of coefficients at different sub-bands. First, they apply a one level DWT to the watermarked image and then on the original image. The difference (corrupted watermark) of the DWT coefficients in the HH band of the watermarked and the original image is then calculated. Then, the cross-correlation between the corrupted watermark and the part of the original watermark that was added in the HH band is determined. If there is a peak in the cross correlation, the watermark is considered to be detected, otherwise they consider the other bands at the same level (i.e. HH and LH, then HH, LH and HL). In case the watermark still cannot be detected, they compute a new level of the DWT and try to detect the watermark again. This process is performed until the watermark is detected or the last level of the DWT has been reached.

Kundur and Hatzinakos [27] embed a binary watermark into the detail wavelet coefficients of the host image with the use of a key. This binary randomly generated key is used to select the exact locations in the wavelet domain (ones location) in which to embed the watermark. First, they compute the $L^{th}$ level discrete wavelet decomposition of the host image to produce a sequence of a $3L$ detailed images. Then, for each level, the embedding modulation at any of the selected coefficients is undertaken as
2.3 Frequency Domain Transforms

follows:

- Order the horizontal, vertical and diagonal detail coefficients at this location (high, middle, and low).

- The range of values between high and low is divided into bins of width \((high - low)/(2Q - 1)\) where \(Q\) is a user-defined variable. These bins represent 1 and -1 in a periodic manner.

- To embed a watermark bit of value 1, the middle coefficients are quantized to the nearest 1 bin. Alternatively, to embed a bit of -1, the middle coefficient is quantized to the nearest -1 bin.

Finally, apply the inverse wavelet transform to form the watermarked image.

In Kunder and Hatzinakos [28] the host is transformed to the \(L^\text{th}\) level discrete wavelet decomposition. Only the first level discrete wavelet decomposition of the watermark is performed. The watermark is a random binary two-dimensional array. It is required that the size of the watermark in relation to the host image be small. The detail images of the host at each resolution level are segmented into a non-overlapping rectangle. Each rectangle has the same size as any bands of the watermark. A numerical measure of perceptual importance (salience) of each of these localized segments is computed. The watermark is embedded by a simple scaled addition of the watermark to the particular detail component. The scaling of the watermark is a function of the salience of the region. The greater the salience, the stronger the presence of the watermark. Finally the corresponding \(L^\text{th}\) level inverse wavelet transform is performed. Salience is computed based on a well-known model given by Dooly [29], which is based on contrast sensitivity.
The original image is required in the extraction process. The extraction process is done by applying the inverse procedure at each resolution level to obtain an estimate of the watermark. The estimates for each resolution level are averaged to produce an overall estimate of the watermark.

Ohnishi and Matsui [30] propose an algorithm similar to the Kunder [27] technique. The most significant difference between the two methods lies in the merging stage of the watermark. Here the author marks the host by forcing the modulo 2 difference between the largest and smallest wavelet coefficients for a particular position and resolution level to be one if \( w(n) = 1 \) and to be zero if \( w(n) = -1 \).

In Barni, et al. [31] the authors present a public watermarking system. A binary pseudo-random sequence is weighted with a function, which takes into account the human visual system (orientation, brightness, and texture) and is then added to the DWT coefficients of the three largest detail sub-bands of the image (i.e., first level). For watermark detection, the correlation between the watermark to be tested for its presence and the marked coefficients is computed. The value of the correlation is compared to a threshold to decide if the watermark is present or not. Experimental results prove the imperceptibility of the watermark and the robustness against the more common attacks. A model for estimating the sensitivity of the eye to noise - previously proposed for compression applications [32] - is used to adapt the watermark strength to the local content of the image.

Inoue, et al. [33] propose two public digital watermarking schemes to embed a binary code. Both methods are built on a data structure called a zerotree, which is defined in the Embedded Zerotree Wavelet (EZW) algorithm of Shapiro [34]. Zerotree coding is based on the hypothesis that if a
wavelet coefficient at a coarse scale is insignificant with respect to a given threshold $T$, then all wavelet coefficients of the same orientation in the same spatial location at a finer scale are likely to be insignificant with respect to $T$. The zerotree is used to classify wavelet coefficients as insignificant or significant as follows: given an amplitude threshold $T$, if a wavelet coefficient $x$ and all of its descendants (i.e., coefficients corresponding to the same spatial locations but at finer scales of similar orientation) satisfy $|x| < T$ then they are called insignificant with respect to a given threshold $T$ or zerotree for the threshold $T$ (otherwise significant coefficients). In one method, the zerotrees are constructed for any coarsest sub-band (except LL sub-band) for a specific threshold. Each watermark binary digit is embedded by writing the same data in the location of all elements of the current zerotree. Data is redundantly embedded because insignificant coefficients are generally easy to change under the influence of common signal processing. In the second method, the watermark can be embedded by thresholding and modifying significant coefficients at the coarser levels. However, it is well known that the modification of these components can lead to perceptual degradation of the signal. To avoid this, they make the value of $T$ larger than in the previous method. As a result, the regions in which the watermark is embedded are applied to detailed portions, i.e., edges or textures, in the coarsest scale component. Therefore, embedding the watermark into significant coefficients is difficult for human eyes to perceive. The watermark is detected by using the position of the zerotree’s root and the threshold value after the wavelet decomposition of the cover image. It is shown that the proposed method is robust against several common signal processes.
2.3.3 Discrete Fourier Transform

The Discrete Fourier Transform (DFT) of a function provides a quantitative picture of the frequency content in terms of magnitude and phase. This is important in a wide range of physical problems and is fundamental to the processing and analysis of signals and images. It is very important to know the properties of DFT so that it can be exploited efficiently. Some of these properties are listed below:

Positive Symmetry If \( f(n,m) \) is real (which is the case of images), its Fourier transform is conjugate symmetric [24]; that is

\[
F(p,q) = F^*(N-p,M-q)
\]

To ensure the inverse DFT is real, changes in the magnitude must preserve positive symmetry.

Negative Symmetry This is the same as the above with regard to the phase component (\( \phi \)), but in this case, we have a negative symmetry compounded in the result

\[
\phi_{p,q} = \phi_{p,q} + \delta
\]

\[
\phi_{N-p,M-q} = \phi_{N-p,M-q} - \delta
\]

Scaling Scaling in the spatial domain causes inverse scaling in the Fourier domain (i.e. as the spatial scale expands, the frequency scale contracts and the amplitude increases vertically in such a way as to keep the area constant), i.e.

if \( f(x_1, x_2) \xrightarrow{DFT} F(k_1, k_2) \) then \( f(ax_1, ax_2) \xrightarrow{DFT} \frac{1}{a} F\left(\frac{k_1}{a}, \frac{k_2}{a}\right) \)
2.3 Frequency Domain Transforms

Translation The translation property of Fourier transform is defined as

\[ f(x_1, x_2) \xrightarrow{DFT} F(k_1, k_2) \]

\[ f(x_1 + a, x_2 + b) \xrightarrow{DFT} F(k_1, k_2)e^{-j(ak_1 + bk_2)} \]

This indicates that the phase is altered only by a translation, i.e. the amplitude is insensitive to the spatial shift of an image. Note that both \( f \) and \( F \) are periodic functions so it is implicitly assumed that the translation causes the image to be 'wrapped around' (circular translation) [35]. By spatial translation (i.e. zero padding) the frequency sampling will be increased.

Rotation Rotating the image through an angle in the spatial domain causes the Fourier representation to be rotated through the same angle [35], i.e.

\[ f(x_1\cos\theta - x_2\sin\theta, x_1\sin\theta + x_2\cos\theta) \xrightarrow{DFT} F(k_1\cos\theta - k_2\sin\theta, k_1\sin\theta + k_2\cos\theta) \]

Log-Polar Representation Most watermarking algorithms have problems in extracting the watermark after an affine geometric transformation on the watermarked object. Some methods try to invert the effect of geometric distortion using the original image. An alternative way is to build a system that can detect the watermark even after a geometric distortion is applied, i.e. Rotation, Scaling and Translation invariance (RST invariant). Most of these systems use the properties of log-polar representation of the spectrum. In a log-polar mapping (which is defined as \( x = e^{r\cos\theta}, y = e^{r\sin\theta} \)), the rotation and scaling in a Cartesian coordinate system will result in a translation in the logarithmic coor-
2.3 Frequency Domain Transforms

dinate system, i.e.

\[
\text{if } f(x, y) \xrightarrow{\text{log-polar-mapping}} f(\mu, \theta) \text{ then }
\]

\[
f(ax, ay) \xrightarrow{\text{log-polar-mapping}} f(\mu + \log a, \theta)
\]

and

\[
\text{if } f(x \cos(\theta + \delta) - y \sin(\theta + \delta), x \sin(\theta + \delta) + y \cos(\theta + \delta)) \xrightarrow{\text{log-polar-mapping}} f(\mu, \theta + \delta)
\]

From the translation property of the Fourier transform as well as the properties of a log-polar mapping, RST invariant domain can be created by applying the Fourier transform to the log-polar version of the Fourier magnitude of an image, which is equivalent to computing the Fourier-Mellin transform.

Phase and Magnitude Modulation The DFT is generally complex valued. This leads to a magnitude and phase representation for the image [36]. The human visual system is far more sensitive to phase distortion than magnitude distortion and as a consequence, the DFT magnitude can be altered significantly without affecting the perceived quality of the image. On the other hand, the phase modulation can possess superior noise immunity when compared to amplitude modulation. As a consequence of this, if a watermark is introduced into the phase components with high redundancy, the attacker would need to cause serious damage to the quality of the image.

Embedding Techniques in the DFT Domain

O’Ruanaidh, et al. [36] investigate the use of DFT phase for the transformation of information. The condition that the image is composed of real
2.3 Frequency Domain Transforms

data implies that the Fourier spectrum is symmetric, and because the human eye is more sensitive to phase distortion, watermarking that changes the phase must preserve the negative symmetry. The most significant components are then selected by comparing the component magnitude squared to the total energy in the spectrum. To detect the watermark, the marked image is simply compared with the original image.

Solachidis and Pitas [37] propose a watermarking method robust to rotation and scaling. The watermark consists of a 2-D circularly symmetric sequence taking values 1, -1. It has zero mean value. The region in which the watermark is embedded should be a ring covering the middle frequencies. The ring is separated in $S$ sectors and in homocentric circles. Each sector is assigned the same value (1, -1). The watermark is added directly to the magnitude of the DFT domain. If the magnitude becomes negative, it is rounded to 0. The 'conjugate symmetry' property for the DFT must be preserved. The original is not required for detection. The detection is done by finding the correlation between the possibly watermarked coefficients and the original watermarks, comparing the correlation against a threshold.

Kim, et al. [38] discuss the embedding of a binary image (seal image) into another image. The entire watermark is modulated by a binary pseudo-noise matrix ($P$). The pseudo noise serves for spreading the watermark evenly and is the secret key for retrieving the watermark. The watermark is embedded into the Fourier domain of the cover image by altering the magnitude components ($m_{ij} = m_{ij} + \alpha \cdot P_{ij} \cdot w_{ij}$). The amplitude factor $\alpha$, is a constant determining the signature strength. The retrieval process can be done without knowledge of the original image. This process starts by approximating the magnitude of the Fourier coefficients of the original image. This can be done by finding the average of the magnitude coefficients around
each point in the watermarked cover. The difference between the predicted and the actual value in the watermarked version is divided by the pseudo-noise that was used in the embedding process (which can be regenerated using the key). Experimental results show that this schema gives a high robustness to the distortion such as blurring and lossy compression.

Raymond, et al. [39] have proposed a modification to the system in [38]. They embed a reduced version of the watermark several times using the same method. This repetition can be used in the retrieval process to enhance the watermark.

Ó Ruanaidh and Pun [35] have introduced the use of the Fourier-Mellin transform for watermarking to embed a watermark which is RST invariant from a digital image. A Fourier transform is first applied which is then followed by a Fourier-Mellin transform. The invariant coefficients are pre-selected for their robustness to image processing and are marked using spread spectrum techniques. The inverse mapping is computed (an inverse log-polar mapping followed by an inverse DFT). Note, that the inverse transformation uses the phase computed during the forward transformations. To extract the watermark, the watermarked image is transformed into the RST invariant domain which then decodes the watermark.

In Herrigel, et al. [40], the embedding process starts by dividing the image into adjacent blocks. The same watermark is embedded in each block. They then map each block into perceptually 'flat' domains by replacing the intensity of each pixel with their logarithm. This step ensures that the intensity of the watermark is diminished in the darker regions of the image where it would otherwise be visible (Weber-Fechner law for HVS response to change of luminance). The Fourier transform is then computed for each
block. Finally, the watermark is modulated with magnitude components selected from the middle-frequency bands. To detect rotation and scaling, a template $T$ is embedded into selected components in log polar space. To determine the rotation and scaling that the image suffered, they calculate the normalized cross correlation between the log-polar components and the template pattern $T$ to find the point of best correlation. If the image has neither been rotated nor scaled, then this point is at the origin.

Lin, et al. [41] propose a watermarking algorithm that is robust to RST distortions. The watermark is embedded using the following steps:

1. Find the discrete log-polar mapping for the Fourier magnitude components of the input image ($M$ rows, $N$ columns).
2. Sum the logs of all values in each column (angle dimension) and add the result of summing column $j$ to the result of summing column $j + N/2$ storing the result in a vector ($v$).
3. Mix the watermark with $v$ using a weighted average of $w$ and $v$ to produce vector $s$.
4. Modify all the values in column $j$ of the log-polar Fourier transform so their logs sum to $s_j$ instead of $v_j$.
5. Invert the log-polar re-sampling of the Fourier magnitude, thus obtaining a modified Cartesian Fourier magnitude.
6. The complex terms of the original Fourier transform are scaled to have the new magnitudes found in the modified Fourier transform. The inverse DFT is applied to obtain the watermarked image.
7. The detection process is as follows: Apply the same signal-extraction process to the watermarked image to produce the extracted vector $v$. Compute the correlation coefficient between $v$ and input watermark vector $w$. If the correlation is greater than a threshold $T$, then the watermark is present, otherwise it is absent.
2.4 Discussion

Spatial domain techniques are in general simpler to implement than methods that place the watermark in the transform domain. In addition they often require a lower computational cost but can be less robust against tampering. Watermarking in the frequency domain possesses a number of desirable features that have been discussed in this chapter which make them superior to the spatial domain embedding techniques. However, each transform space has its own advantages in a particular application. For example, embedding in DCT domain is simple and straightforward and is compatible with JPEG compression, DWT can be used efficiently with HVS, and DFT is relatively immune to geometric distortion.

Most of the watermarking techniques embed the watermark by adding a pseudo-random noise field, spreading a small image over a much larger image, or thresholding frequency components. However, most of these methods are not optimized for printed works, and may have difficulties in extracting the watermark after a printing and scanning process. One important observation about these techniques is that they place no emphasis on and take no account of the physical process which captured the image in the first place. The techniques developed in this thesis begin with the assumption that if the watermark embedding process is founded on the basic principles of an optical system then the result will be a watermarking system that is robust to the scanning process.

In next chapter, the physical principles of an imaging system will be derived in order to produce an image encryption and watermarking model that is compatible with image capture devices.
Chapter 3

Optical Image Formation

This chapter explores how images are formed during the process of scanning a document. It is critical - to this thesis - that the physical principles of an imaging system and the theory of image formation are considered in order to derive an image encryption and watermarking model that is compatible with image capture devices (digital cameras and scanners for example). This is one of the underlying themes of what is called 'Image Understanding'. Image processing techniques require some form of knowledge concerning the physical system that provides the image. Describing the process and equations of image formation is motivated by the desire to provide methods of image restoration and reconstruction.

This chapter begins by investigating how digital images are captured, and goes on to study the image-forming properties of optical systems as a whole. This leads directly to an investigation of the principles of what is commonly known as 'Fourier Optics'. These principles lead to the basic convolution model for an optical imaging system and the concept of optical image formation in general, from which specific models for coherent and
incoherent image formation can be derived. The basis of diffraction theory is also considered, with a focus on analytical techniques that are fundamental to modelling most imaging systems. Finally, the Fourier transforming properties of the lens are investigated.

3.1 Digital Image Capturing

Every visual scene is an image, the images are formed upon the human retina by the iris-lens portion of the human eye and thus, the eye embodies an image formation system (iris-lens) and image sensor or recording system (retina) [42]. Images may be formed and recorded as digital images using the same principles as that of the human eye (i.e. an image formation and image sensor system). Understanding the effects of an image formation and sensing system upon the image must therefore be understood and calibrated.

The types of images in which we are interested in are generated by the combination of a source of 'illumination' and the reflection or transmission (with absorption) of energy from that source by the elements of the scene (object) being imaged [24]. The object is either a source of energy (passive imaging systems) or it is illuminated by a source of energy (active imaging systems).

Depending on the nature of the source, illumination energy will be reflected, transmitted (or emitted) by the object which then propagates through space. An example of the first category is light reflected from a planar surface. An example of the second category is when X-rays pass through an object for the purpose of generating a diagnostic X-ray film. Most of the light that reaches our eyes is reflected off objects to produce the image that
3.2 Image Formation Systems

we see. An image formation system, intercepts the propagating energy and transforms it to a form that can be detected by sensor devices. The energy received at the sensor side is transformed into a voltage by a combination of input electrical power and sensor material that is responsive to the particular type of energy being detected. The output voltage waveform is the response of the sensor(s); a digital quantity being obtained from each sensor by digitizing its response. One sensor geometry - that is used in most flatbed scanners - consists of an in-line arrangement of sensors in the form of a sensor strip. The strip provides imaging elements in one direction. Motion perpendicular to the strip provides imaging in the other direction [24]. In summary, digital images are recordings of energy (light) emitted from a particular scene and stored in computer memory.

3.2 Image Formation Systems

Even though the optical imaging system may be complex, the process of image formation can be simplified by adopting a 'black box' approach to the problem. Consider an optical imaging system as a 'black box', with an input wavefront from the object and an output wavefront forming the image. If we know how the wavefront is altered in a given direction, then we can predict how the image will be formed. This approach focuses on the image-forming properties of the optical system as a whole, rather than on analyzing the physical and optical properties of every component in the system.

As shown in Figure 3.1, \( f(\xi, \eta) \) is considered as an object in the coordinate system \((\xi, \eta)\), that is referred to as the object plane. In an image formation system, the rectangular box in Figure 3.1, intercepts the propa-
gating energy and transforms it to an image in the coordinate system \( (x, y) \) which is referred to as the image plane [42]

The laws of physics and experimental measurement show that lens- and aperture-based optical systems (like the human eye and most cameras) act as linear systems that transforms the two-dimensional light distribution in the object plane into a two-dimensional light distribution on the imaging surface. For example, increasing the intensity of the point source at the object plane causes a proportional increase in the intensity of the corresponding points in the image plane. In other words, all of the optical processing between \( f(\xi, \eta) \) (Figure 3.1) and \( f(x, y) \) behaves as a linear system.

Optical systems produce two effects on an image - projection and degradation. The degradation is due to:

1. **Aberrations**, which cause the exit wave to depart from its ideal spherical shape. The wave aberration defines how the phase of the light field is affected as it passes through the optical system. In aberration-free systems, a core of light emanating from any point in a small patch within the object plane, when captured by the optical system, is turned into
3.2 Image Formation Systems

a convergent cone that comes to focus at a point in the image plane. Optical systems can be designed with minimal aberration

2 Diffraction, classically, light is thought of as travelling in straight lines, but in reality, light waves tend to bend around small obstacles, nearby barriers and spread out past small openings or slits. For example, when light from a point source passes through a small circular aperture, it does not produce a bright dot as an image, but rather a diffuse circular disc known as an Airy disc surrounded by much fainter concentric circular rings. This phenomenon is known as diffraction. This example of diffraction is of significance because the eye and many optical instruments have circular apertures. Due to the wave nature of light and finite aperture based optical systems, diffraction cannot be avoided or reduced. Understanding the effects of diffraction upon the image must therefore be understood. High quality, aberration-free systems are called diffraction-limited systems.

The output of an optical system also depends on the nature of the input illumination. There are two cases:

1. Coherent illumination. When the object illumination (waves originating from one common origin) is such that the field is varying in perfectly predictable way across all points in the object plane then this is referred to as coherent illumination. If the point sources being imaged vary in synchrony, then stable patterns of constructive and destructive interference exist (such as spherical waves). The coherence assumptions are seldom satisfied in optical image formation systems that are encountered in daily life.
2. Incoherent illumination. Here, when there are more than one independent light sources illuminating the object plane, the physical condition at those points is called incoherence (i.e. varies randomly from one point to another).

Finally, the medium in which the observation is made is critical. In this discussion, it is assumed to be homogeneous.

### 3.3 Fourier Optics

The use of linear-systems techniques to measure and describe optical systems is called **Fourier optics** (as opposed to geometrical optics). These techniques are useful for understanding optical systems where diffraction effects are significant.

#### 3.3.1 Linear System Properties

A system may be defined as that which produces a set of output functions from a set of input functions. Physically, it may be an electrical circuit (inputs and outputs are voltages) or an optical system where the inputs and outputs are either complex amplitudes or intensities. From the point of view of linear systems theory, the physical nature of the system is unimportant. An arbitrary 'system' can be represented by the operator $\hat{L}$ such that

$$s(x, y) = \hat{L}[f(x, y)]$$

A linear system has the property that

$$\hat{L}[af_1(x, y) + bf_2(x, y)] = a\hat{L}[f_1(x, y)] + b\hat{L}[f_2(x, y)]$$
for all inputs \( f_1 \) and \( f_2 \) and all constants \( a \) and \( b \). Linearity implies that an output function can be broken down into elementary functions, each of which can be separately passed through the system; the total output is then the sum of the 'elementary' outputs.

The 'shifting property' of the delta function allows us to consider any input function to be a linear combination of weighted and displaced delta functions, i.e.

\[
f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x', y') \delta(x - x') \delta(y - y') \, dx' \, dy'
\]

giving an output

\[
s(x, y) = \hat{L}[f(x, y)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x', y') \hat{L}[\delta(x - x') \delta(y - y')] \, dx' \, dy'
\]

The system response at \( (x, y) \) due to a delta function input at \( (x', y') \) is called the Impulse Response Function (IRF). Let

\[
p(x, y; x', y') = \hat{L}[\delta(x - x') \delta(y - y')]
\]

Then, for a linear system,

\[
s(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x', y') p(x, y; x', y') \, dx' \, dy'
\]

If the impulse response function of a linear system depends only on the coordinate differences \( (x - x') \) and \( (y - y') \), and not on each coordinate separately, i.e.

\[
p(x, y, x', y') \equiv p(x - x', y - y')
\]

then an expression for \( s \) is obtained which involves the simple convolution relationship

\[
s(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x', y') p(x - x', y - y') \, dx' \, dy'
\]
3.3 Fourier Optics

Such a system is called a stationary and an optical system with this property is known as isoplanatic (refer to Section 3.3.2 for more details).

The convolution relationship between input and output suggests using Fourier transform techniques which, via the convolution theorem, gives

$$ S(u, v) = F(u, v)P(u, v) $$

where

$$ P(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y) \exp[-2\pi i(ux + vy)]dx dy $$

with a similar relationship between $s$ and $S$ as $p$ and $P$. The quantity $P$ is called the system transfer function. Note, that the convolution relationship only applies to linear stationary systems. For non-stationary systems, there is no equivalent convolution theorem.

3.3.2 Optical Systems as Linear Systems

The image formation system is a two-dimensional linear system. This means that point sources in the object plane are combined in image plane by addition. The corresponding properties of an optical system are then as follows.

Point Spread Function (PSF)

In optical imaging systems, the Impulse Response Function is called the Point Spread Function (PSF). The PSF is an important measure of image quality. It expresses the way the imaging system smears or spreads a single point of light. The image produced by a point-source is called the point spread function. In the case of a diffraction limited (no aberrations) system, the PSF is the Airy disk. Aberrations will cause additional spreading or blurring.
of the point. In other words, if any scene imaged by an optical system is considered to be a collection of points, the image of each point in the scene will be the PSF, and the integration of all the points will build the image of the scene. Mathematically, the image (after projection) can be described as a convolution of the object with the PSF of the system.

All image formation systems are inherently resolution limited. Note, that as the aperture size increases, the PSF becomes narrower. This allows objects to be imaged with higher resolution, and is (part of) the reason for telescopes having such a large diameter.

**Shift Invariance**

Stationary optical systems are isoplanatic. Isoplanicity requires that the point spread function is the same for all wavefield angles and implies that the aberrations are independent of wavefield angle. For reasonably small off-axis distances in good optical systems, the shape of the spot undergoes essentially no change. Thus, the system can be assumed to be shift invariant (or isoplanatic).

**Optical Transfer Function (OTF)**

The counterpart in the frequency domain to the PSF is the Optical Transfer Function (OTF), which is the Fourier transform of the PSF. The OTF is a complex number whose magnitude is the modulation transfer function (MTF), and the phase is the phase transfer function (PTF), i.e.

$$ OTF = MTF * e^{i \Phi_{PTF}} $$
The modulation transfer function, describes the transfer or modulation of sinusoidal components of the object. The PTF describes spatial shifts of the sinusoidal components.

The Pupil Function

The way light is changed by the optical system can be represented by the complex pupil function

\[ P(x, y) = A(x, y)e^{i(2\pi/\lambda)W(x, y)} \]

where \( \lambda \) is the wavelength. The pupil function has two components:

1. the amplitude component \( A(x, y) \);
2. the phase component which contains the wave aberration \( W(x, y) \).

The amplitude component (absorption term) \( A(x, y) \) defines the shape, size and transmission of the optical system. The most common shape for the aperture function is a circ function that defines a circular aperture. The size is simply the size of the pupil that the pupil function defines.

The wave aberration \( W(x, y) \) defines how the phase of light is affected as it passes through the optical system. If the wavefront aberration function, \( W(x, y) \) is known, the pupil function, \( P(x, y) \), can be calculated.

A clear optical system has a value of unity at all points across the pupil function. For a circular lens of radius \( a \),

\[
P(x, y) = \begin{cases} 
1 & x^2 + y^2 \leq a^2 \\
0, & \text{otherwise}
\end{cases}
\]
3.3 Fourier Optics

The OTF (of incoherent light) may be computed by convolving the pupil function with its complex conjugate, \( P^* \). This is the same as taking the autocorrelation of the pupil function. By the convolution theorem, this operation can be performed in the frequency domain by multiplication. Therefore, the Fourier transform of the OTF is the product the Fourier transforms of the pupil function and its complex conjugate.

**Linear and Shift Invariant System (LSI)**

Many real imaging systems are (to a good approximation) both linear and isoplanatic. Therefore, it can be completely described by either the PSF or OTF.

The mathematics of image formation begins with the computation of the point spread function (PSF). Consider the case where the object plane is illuminated by a plane or spherical wave. Let the complex amplitude immediately after the object be denoted by \( U_m(x, y) \) and \( U_{\text{out}}(x, y) \) be the complex amplitude at the image plane and let the complex amplitude at \((x, y)\) in the output due to a unit strength point at in the input be \( p(x, y; x', y') \).

The total amplitude at \((x, y)\) due to all such points in the object plane is then given by

\[
U_{\text{out}}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U_m(x', y') p(x, y; x', y') dx' dy'
\]

For an isoplanatic optical system, this reduces to

\[
U_{\text{out}}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U_m(x', y') p(x - x', y - y') dx' dy'
\]

What we actually observe (in the image plane) with the eye is the intensity
I, so we are really viewing the power spectrum of the $U_{\text{out}}(x,y)$ object. The intensity is given by $|U_{\text{out}}(x,y)|^2$.

This analysis assumes that the object is illuminated by perfectly spatially coherent light. Coherent illumination implies that $U_{\text{in}}$ or $U_{\text{out}}$ does not vary in time. On the other hand, incoherent illumination varies in time. This makes the analysis associated with coherent light different to that of incoherent light. The following section investigates the effect of the illumination on the computation of the PSF and OTF.

### 3.4 Illumination and Image Formation

#### 3.4.1 Incoherent Image Formation

Consider the case of narrowband light that is not perfectly spatially coherent. The general complex representation of the time-varying scalar field is called the analytic signal $V(r,t)$; it is defined such that

$$\text{real scalar field} = \Re[V(r,t)]$$

For narrowband light, the analytic signal can be written as a product of a 'slowly varying' function - the time-varying complex amplitude $U(r,t)$ times $\exp(-i\omega t)$, i.e.

$$V(r,t) = U(r,t) \exp(-i\omega t)$$

where $U$ is the scalar complex amplitude, $\omega$ is the angular frequency ($=2\pi \times$ frequency), $t$ is time and $r$ is a three-dimensional vector which in Cartesian coordinates is given by

$$r = xx + yy + zz$$
The instantaneous intensity is defined as
\[ I(r, t) = |U(r, t)|^2 \]
whereas the time-averaged intensity \( \bar{I}(r) \) (i.e., that observed by an optical detector), is equal to
\[ \bar{I}(r) = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} I(r, t) dt \]
In general, the time-varying complex amplitudes are related by
\[ U_{\text{out}}(x, y, t) = \int \int U_{\text{in}}(x', y', t) p(x, y, x', y') dx' dy' \]
The average intensity is therefore given by
\[ \bar{I}_{\text{out}}(x, y) = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} |U_{\text{out}}(x, y, t)|^2 dt = \int \int_{-\infty}^{\infty} p(x, y; x', y') p^*(x, y, x', y') \times \left[ \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} U_{\text{in}}(x', y', t) U_{\text{in}}^*(x'', y'', t) dt \right] dx' dy' dx'' dy'' \]
The term in [ ] is called the mutual intensity of the narrowband light
\[ J_{m}(x', y'; x'', y'') = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} U_{\text{in}}(x', y', t) U_{\text{in}}^*(x'', y'', t) dt \]
or
\[ J_{m}(r', r'') = \langle U_{\text{in}}(r', t) U_{\text{in}}^*(r'', t) \rangle \]
In order to simplify the above expression for \( \bar{I}_{\text{out}}(x, y) \) incoherent light can be defined as:
\[ J(r', r'') = \bar{I}(r') \delta(r' - r'') \]
This means that two neighbouring points \( r' \) and \( r'' \) have uncorrelated fields, for any \( r' \neq r'' \). Using the definition for incoherent light above, the expression for \( I_{\text{out}} \) becomes

\[
I_{\text{out}}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y; x', y')p^*(x, y; x', y')
\times \bar{I}(x', y')\delta(x' - x'')\delta(y' - y'')dx'dy'dx''dy''
\]

or

\[
I_{\text{out}}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left| p(x, y; x', y') \right|^2 \bar{I}(x', y')dx'dy'
\]

The quantity \( \left| p(x, y; x', y') \right|^2 \) is the intensity point spread function. For an isoplanatic optical system, this result reduces to

\[
I_{\text{out}}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \bar{I}(x', y') | p(x - x', y - y') |^2 dx'dy'
\]

where the bar over \( I \) has been omitted when referring to the intensity - a time average is always assumed.

For perfectly incoherent illumination, an optical system is linear in intensity and, if isoplanicity holds, the output (image) intensity is equal to the input (object) intensity convolved with the intensity point spread function.

### The OTF and Pupil Function

The incoherent optical transfer function (IOTF) \( T(u, v) \) is the Fourier transform of the point spread function. Applying the autocorrelation theorem to the PSF,

\[
T(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P(\xi, \eta)P^*(\xi + \lambda u, \eta + \lambda v)d\xi d\eta
\]
3.4 Illumination and Image Formation

where $P(\xi, \eta)$ is the inverse Fourier transform of $p(x, y)$,

$$P(\xi, \eta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y) \exp \left[ \frac{-2\pi i}{\lambda z} (\xi x + \eta y) \right] dx \, dy$$

and is the pupil function. The equation above for $T$ basically states that the IOTF is equal to the spatial autocorrelation of the pupil function. Note, that the IOTF relates the input and output intensity spectra

$$\tilde{I}_{\text{out}}(u, v) = \tilde{I}_{\text{in}}(u, v) T(u, v)$$

The spatial frequencies are intensity frequencies and are not the same as the amplitude frequencies produced in a coherent optical system.

3.4.2 Coherent Image Formation

Coherent illumination implies that $U(x, y, t) = U(x, y)$ does not vary in time. Suppose that the object plane is illuminated by a plane or spherical wave - by perfectly spatially coherent light. The complex amplitude of the image is then equal to that in the object plane convolved with the amplitude point spread function (for an isoplanatic system), i.e.

$$U_{\text{out}}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U_{\text{in}}(x', y') p(x - x', y - y') dx' \, dy'$$

where

$$p(x, y) = C \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P(u, v) \exp \left[ -\frac{2\pi i}{\lambda z} (ux + vy) \right] du \, dv$$

and $P$ is the pupil function of the optical system, i.e. the complex amplitude in the exit pupil. The constant $C$ is usually chosen so that

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y) dx \, dy = P(0, 0) = 1$$
3.4 Illumination and Image Formation

Taking the Fourier transform of \( U_{\text{out}} \) and using the convolution theorem we can write

\[
\tilde{U}_{\text{out}}(u, v) = \tilde{U}_{\text{in}}(u, v)T(u, v)
\]

where \( \tilde{U}_{\text{out}} \) is the spectrum of image amplitude, \( \tilde{U}_{\text{in}} \) is the spectrum of object amplitude and \( T \) is the coherent optical transfer function (COTF). Note that

\[
T(u, v) = \iint_{-\infty}^{\infty} p(x, y) \exp[-2\pi i (ux + vy)] \, dx \, dy = P(\lambda u, \lambda v)
\]

i.e. the COTF at spatial frequency \((u, v)\) is simply equal to the pupil function at coordinates \((\lambda u, \lambda v)\)

The intensity distribution \( I(x, y) \) of a coherent image is given by,

\[
I(x, y) = |U_{\text{out}}|^2 = |p(x, y) \otimes \otimes U_{\text{in}}(x, y)|^2
\]

The model can be used to develop computer generated speckle patterns for example where \( U_{\text{in}}(x, y) \) is constructed using white Gaussian noise. Physically the noise function represents a first approximation to a (rough) surface from which coherent light is scattered.

### 3.4.3 Summary

Table 3.1 gives a summary of the mathematics related to incoherent and coherent image formation (in which isoplanaticity is assumed)

To illustrate image formation in terms of the previous mathematical analysis, the image formation process for several systems -that are in common use- are discussed next
3.5 Diffraction-based Optical Systems

Diffraction theory is the study of light propagation phenomenon. When a wave, such as light, passes through a small aperture, it will be distorted. It will form a distinctive pattern on a screen, known as the diffraction pattern. This pattern contains information on the diffracting aperture.

The phenomenon of diffraction is common to 1D transverse (and hence scalar) waves such as water waves, true scalar waves such as in acoustics and vector waves as in optics. All three cases appear to exhibit the same magnitude of diffraction phenomena.

Figure 3.2 illustrates the geometry of the simplest optical system. The object plane is an opaque sheet except for an opening aperture. This plane is illuminated from the left by a uniform plane wave. Only the aperture passes the wave. The wave propagates through a homogeneous medium to the screen (image plane). Our task is to determine the field amplitude and phase at some observation position behind the aperture. The field at the observation point is found by integrating the contributions from each source.

### Table 3.1: Principal properties of coherent and incoherent image formation

<table>
<thead>
<tr>
<th>Property</th>
<th>Incoherence</th>
<th>Coherence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linearity</td>
<td>Intensity</td>
<td>Complex amplitude</td>
</tr>
<tr>
<td>PSF (p)</td>
<td>(</td>
<td>p(x, y)</td>
</tr>
<tr>
<td></td>
<td>scaled (</td>
<td>\text{FFT}(P)</td>
</tr>
<tr>
<td>OTF ((u, v))</td>
<td>(P \otimes P)</td>
<td>(P(\lambda u, \lambda v))</td>
</tr>
<tr>
<td>Pupil ((P))</td>
<td>(A(x, y)e^{(2\pi/\lambda)W(x, y)})</td>
<td>(A(x, y)e^{(2\pi/\lambda)W(x, y)})</td>
</tr>
<tr>
<td>(I_{\text{out}}(x, y))</td>
<td>(</td>
<td>U_{\text{in}}(x, y, t)</td>
</tr>
</tbody>
</table>
of propagation.

Since the propagation is in free space, the wave equation in free space can be used to determine the image distribution. The solution of the wave equation, given the boundary conditions imposed by the aperture and assuming that the aperture does not alter the illumination field in its open area, is known as (Kirchhoff) diffraction theory (for further details see Appendix A.1).

Solutions of the wave equation are not simple. There are two solutions, however, that can be derived by approximation, Fraunhofer diffraction and Fresnel diffraction.

### 3.5.1 Fraunhofer Diffraction

Fraunhofer diffraction deals with the limiting case where the light approaching the diffracting object is parallel and monochromatic (light that is void of colour is called achromatic or monochromatic light). The only attribute
3.5 Diffraction-based Optical Systems

of such light is its intensity and assumes that the diffracted wavefield is observed a large distance (compared to the size of the diffracting object) from the screen - the point of observation is in the far field. For this reason, Fraunhofer diffraction is sometimes called diffraction in the 'far field'.

Under the assumption of coherent illumination, the complex field distribution in the image plane is given by [43] (a detailed description is included in Section A.3)

\[
U(x_0, y_0) = \frac{i \exp(ikz_0)}{\lambda} \frac{\exp \left( ik \left( \frac{x^2}{2z_0} + \frac{y^2}{2z_0} \right) \right)}{z_0} \int_{-\infty}^{\infty} f(x, y) \exp \left( -\frac{ik}{z_0} (xx_0 + yy_0) \right) \, dx \, dy
\]

where \( z_0 \) is the distance from the aperture to the screen, \( f \) can be taken to describe the 'shape' of the aperture, and \( k = \frac{2\pi}{\lambda} \). Here, \( k \) is the wavenumber and \( \lambda \) is the wavelength.

Apart from the factors in front of the integral, the electrical field distribution in the image plane \( U \) for constant \( z_0 \) is thus established by a Fourier transform of the field strength distribution in the diffracting plane \( f \) evaluated at the coordinates

\[
u = \frac{x_0}{z_0 \lambda} \quad \text{and} \quad v = \frac{y_0}{z_0 \lambda}
\]

As a linear system, the point spread function for Fraunhofer diffraction is

\[
p(x, y; x', y') = \frac{i \exp(ikz)}{\lambda} \frac{\exp \left( ik \left( \frac{x^2}{2z} + \frac{y^2}{2z} \right) \right)}{z} \exp \left( -\frac{ik}{z} (xx' + yy') \right)
\]

What we actually observe with the eye is the intensity \( (I) \), so we are really viewing the power spectrum of the input object. The intensity is given by \( | U |^2 \).

We can therefore write

\[
I(x_0, y_0) = \frac{1}{\lambda^2 z_0^2} | \hat{F}[f(x, y)] |^2
\]

where \( \hat{F} \) is the Fourier transform operator. As a result, the phase factor drops out of the operation.
3.5 Diffraction-based Optical Systems

This result is the Fraunhofer diffraction formula and is based on the following assumptions:

(i) The aperture is thin (compared with the wavelength of light).

(ii) The diffraction pattern is observed in the far field.

3.5.2 Fresnel Diffraction

The more general case, in which the restrictions above are relaxed is called Fresnel diffraction. If the distance from the aperture to screen is small, then the diffraction pattern on the screen is characteristic of Fresnel diffraction and is said to be in the intermediate field or Fresnel zone.

Under the assumption of coherent illumination, the complex field distribution in the image plane is then given by [43] (a detail description is included in Section A.4)

$$U(x_0, y_0) = \frac{i \exp(ikz_0)}{\lambda} \exp \left( ik \frac{x_0^2 + y_0^2}{2z_0} \right)$$

$$\times \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp \left[ -ik \frac{x_0}{z_0} (xx_0 + yy_0) \right] \exp \left[ \frac{ik}{2z_0} (x^2 + y^2) \right] \, dx \, dy$$

Apart from the factors in front of this integral, $U$ is simply equal to the 2D Fourier transform of the function

$$f(x, y) \exp \left[ \frac{ik}{2z_0} (x^2 + y^2) \right]$$

The term

$$\exp \left[ \frac{ik}{2z_0} (x^2 + y^2) \right]$$

is a quadratic approximation to a spherical wave. In other words

$$U \sim \text{spherical wave} \times \hat{F}[f \times \text{spherical wave}]$$

Noting that

$$\frac{ik}{2z_0} (x_0^2 + y_0^2) + \frac{ik}{z_0} (-xx_0 - yy_0) + \frac{ik}{2z_0} (x^2 + y^2)$$
3.5 Diffraction-based Optical Systems

\[ \begin{aligned}
&= \frac{ik}{2z_0} [x_0^2 - 2x x_0 + x^2 + y_0^2 - 2y y_0 + y^2] \\
&= \frac{ik}{2z_0} [(x_0 - x)^2 + (y_0 - y)^2]
\end{aligned} \]

the Fresnel diffraction formula can be written as

\[ U(x_0, y_0) = \frac{i \exp(ikz_0)}{\lambda x_0} \int_{-\infty}^{\infty} f(x, y) \exp \left( \frac{i \pi}{\lambda z_0} [(x_0 - x)^2 + (y_0 - y)^2] \right) dxdy \]

From the last equation we see that \( U \) is essentially (ignoring scaling constants) given by the convolution of the function

\[ f(x, y) \]

with

\[ \exp \left( \frac{ik}{2z_0} [x^2 + y^2] \right) \]

or

\[ U(x, y) = \frac{i \exp(ikz_0)}{\lambda x_0} f(x, y) \otimes \exp \left( \frac{ik}{2z_0} [x^2 + y^2] \right) \]

where \( \otimes \) denotes the 2-D convolution integral. The point spread function for Fresnel diffraction is

\[ p(x, y; x', y') = \frac{i \exp(ikz)}{\lambda z} \exp \left( \frac{i \pi}{\lambda z} [(x - x')^2 + (y - y')^2] \right) \]

Summary

Any light wave of finite spatial extent undergoes diffraction as it propagates through space. If at a certain position \( z_1 \), there is a transparency with an amplitude transmittance described by \( T \), at a position \( z_2 \) far from \( z_1 \) the amplitude variation \( U \) is given by the Fraunhofer approximation, where \( U \) is almost given by a Fourier transform of \( T \). This approximation is valid only for distances far from \( z_1 \). If the observation point is near to the transparency, then the amplitude variation \( U \) is given by Fresnel approximation, where \( U \) is almost given by a convolution of \( T \) with a quadratic approximation to a spherical wave.
3.6 Lens-based Optical Systems

Diffraction limited image formation systems are relatively simple. More realistic image formation systems, such as illustrated in Figure 3.3, contain entrance and exit apertures (entrance pupil and exit pupil) and a lens (or lenses). In this section, the Fourier transforming properties of a lens are investigated, how lenses can be used to form images, and how lenses can be used to filter spatial frequency components.

3.6.1 The Fourier Transforming Properties of Lenses

A simple ideal lens images a point at infinity at its focus. The diffraction pattern can be seen in the back focal plane of the lens. The lens must be 'ideal', i.e. free from all aberrations. This means that the path lengths through the various regions of a lens must not depart from their ideal values by more than a very small fraction of a wavelength. A lens of this perfection is called a diffraction-limited lens. Fourier optics is concerned with the applications which arise as a consequence.
3.6 Lens-based Optical Systems

of the Fourier transforming properties of a lens.

If a source distribution occurs in the form of a collimated beam of light, the 2D Fourier transform of that distribution will occur at the focus of the lens. Thus the amplitude variation is given by Fraunhofer approximation

\[
U(x_0, y_0) = \frac{1}{\lambda} \frac{\exp(ikf)}{f} \exp \left( ik \frac{x_0^2 + y_0^2}{2f} \right) \hat{F}[t(x, y)]
\]

where \( f \) is the focal length of the lens and \( t(x, y) \) is the input distribution. The amplitude is therefore almost given by a Fourier transform - there is an additional quadratic phase factor.

It is interesting to ask whether a transparency can be placed a distance \( d \) in front of the lens so as to give an exact Fourier transform? The amplitude \( T(x, y) \) generated by the transparency in front of the lens is given by the Fresnel diffraction formula

\[
T(x, y) = \frac{1}{\lambda} \frac{\exp(ikd)}{d} t(x, y) \odot \exp \left( \frac{2\pi}{\lambda d} (x^2 + y^2) \right)
\]

and a lens performs a Fourier transform on the field \( T \). Therefore in the focal plane

\[
U(x_0, y_0) = \frac{1}{\lambda} \frac{\exp(ikf)}{f} \exp \left( ik \frac{x_0^2 + y_0^2}{2f} \right) \hat{F}[T(x, y)]
\]

Using the convolution theorem and noting that

\[
\hat{F} \left( \exp \left[ \frac{2\pi}{\lambda d} (x^2 + y^2) \right] \right) = \lambda d \exp[-i\pi \lambda d (u^2 + v^2)]
\]

we have

\[
\hat{F}[T(x, y)] = \tilde{T}(u, v) = \frac{1}{\lambda} \exp(ikd) \tilde{t}(u, v) \exp[-i\pi \lambda d (u^2 + v^2)]
\]

where \( \tilde{t} = \hat{F}[t] \) and

\[
u = \frac{x_0}{x_0 \lambda} \quad \text{and} \quad v = \frac{y_0}{y_0 \lambda}
\]
3.6 Lens-based Optical Systems

The field $U$ is therefore given by

$$ U(x_0, y_0) = \frac{1}{\lambda f} \exp[ik(f + d)] \exp \left[ \frac{i\pi}{f\lambda} (x_0^2 + y_0^2) \right] \times \exp \left[ -i\pi \frac{\lambda d}{(\lambda f)^2} (x_0^2 + y_0^2) \right] \tilde{t}(u, v) $$

$$ = \frac{1}{\lambda f} \exp[ik(f + d)] \exp \left[ \frac{\pi}{\lambda f} (x_0^2 + y_0^2) \left( 1 - \frac{d}{f} \right) \right] \tilde{t}(u, v) $$

Now, when $d = f$

$$ U(x_0, y_0) = \frac{1}{\lambda f} \exp[ik(f + d)] \tilde{t}(u, v) $$

or ignoring scaling constants, when $d = f$

$$ U(x, y) = \mathcal{F}[t(x, y)] $$

Thus, there is an exact Fourier transform relationship between the front and back focal planes of a well-corrected lens system. This analysis ignores the finite extent of the lens, which causes vignetting.

3.6.2 Forming Images using Lenses

A lens can be used to form an image onto the screen. The Fourier transforming properties of a lens are invariant of the direction in which light propagates through the lens. If a mirror is placed at the focal plane of a lens, the Fourier transform of the source distribution will be reflected back through the lens to reproduce the original distribution. Alternatively, the Fourier transform of a source produced at the focus of lens can be reconstructed by employing a second lens. The object should be placed one focal length before the first lens and the image should appear one focal length beyond the second lens, where the separation between the two lenses should be the sum of their focal lengths (see Figure 3.4).
3.6 Lens-based Optical Systems

3.6.3 Optical Filtering

Using two lenses provides a system which can be used to filter the spatial frequency components of a source distribution via the application of certain optical masks in the focal plane of the first lens - spatial filtering. For example, spatial filtering can restore the quality of a collimated laser beam by blocking all spatial frequencies due to the interaction of the beam with dust particles. In general, any 'noise' induced by the (multiple) scattering of light with a complex of subwavelength objects (such as dust particles) modifies the spatial frequencies with high values and so can be removed by application of an optical lowpass filter (a mask placed at the focal plane of a lens).

In general, the use of a lens system to generate the Fourier transform of an image (forward Fourier transform) and to recover an image from this transform (inverse Fourier transform) provides an optical method of processing signals and images. This is known as optical signal processing. Optical signal processing
exploits the Fourier transforming properties of a lens to process (optically filter) an image in the same way that a fast Fourier transform can be used to process (digitally filter) a digital signal/image.

3.7 Deconvolution

The way that incoherent imaging systems produce an image is a combination of three elements: the actual object, the optical qualities of the imaging system and random noise generated for example by imperfections in the detector. This combination can be represented by the equation

\[ I(x, y) = O(x, y) \otimes p(x, y) + n(x, y) \]

where \( I(x, y) \) is the light intensity for the image formed by the optical system, \( O(x, y) \) is the light intensity distribution of the perfect image that would be predicted by the optical system, i.e. without aberrations or diffraction, \( n(x, y) \) represents the noise and \( \otimes \) represents the convolution operation. The most important component of this equation is \( p(x, y) \), the point spread function. This result is the same as that discussed previously (ignoring the noise).

When an image has been degraded in this manner, a number of digital image processing techniques can be employed to 'de-blur' or deconvolve the image and enhance its information content. Nearly all of these techniques are either directly or indirectly based on a mathematical model for the blurred image which involves the convolution of two functions - the Point Spread Function and the Object Function. Hence, 'de-blurring' an image amounts to solving the inverse problem posed by this model which is known as 'Deconvolution'.

By applying a Fourier transform to the convolution equation above, the convolution can be converted into an ordinary product in the Fourier domain. One can take the Fourier transform of the image, divide by the Fourier transform of
3.8 Conclusion

the point spread function or OTF and then transform back again to image space. Division with complex numbers is accomplished by dividing the magnitudes and subtracting the phases. A practical application of this process is the recovery of non-aberrated images obtained through aberrated optics. Using deconvolution or the corresponding operations in the frequency domain, it is theoretically possible to recover high resolution images if the system's OTF is known. In practice, it not always appropriate to just 'divide' through by the optical transfer function, which for many frequencies may have values equal to or near to zero. The practical effect of dividing by near-zero values is that any noise in the system is amplified, and the reconstructed image obtained is noise-dominated.

3.8 Conclusion

The aim of this research is to develop a watermarking system that can operate using standard off-the-shelf low resolution image capture devices such as a digital camera (WebCam) and/or a flatbed scanner. This chapter provides the physical principles of an imaging system and the theory of image formation in order to derive a watermarking scheme that is compatible with the principles of the optical system used.

These hardware devices use incoherent white light to generate an image. Moreover, the formation of an optical image (from a pin hole camera to an integrated lens system) can be based on either the Fraunhofer or Fresnel approximation as discussed in this chapter.

The fundamental result of this chapter, and one that is central to the research in this thesis is the convolution model for an optical imaging system and that fact that there is a Fourier based relationship between the object plane and the imaging plane in the Fraunhofer zone and a convolution relationship in in the Fresnel zone. This result directs us toward a watermarking technique that makes
use of a diffusion process based on a convolution operation which will consequently provide compatibility with the process of optical imaging. This compatibility is essential for an approach that is based on using a hard-copy document that needs to be 'read' using an image capture device.

In the next chapter, and throughout the rest of the thesis, the new approach for print security is introduced and demonstrated.
Chapter 4

Coding Model for Low Resolution Watermarking

4.1 Problem Definition

Paper-based high-value documents have the potential to be counterfeited. Valuable documents have traditionally been provided with security features based on novel printing techniques (e.g. micro-printing) and other overt (e.g. foil holograms) and covert (e.g. printing using ultraviolet sensitive inks) features. This is the result of a continual race between the issuers of valuable documents and those who try to forge or counterfeit them.

Today, all designers use computers to design valuable documents (original works). Some of these works may remain in a digital form, but many will be printed for distribution. Counterfeiters may try to obtain a perfect copy from an original printed version. Since they only have a printed version to work from (i.e. they will normally not have access to the e-copy of the original version), they are likely to use scan/print technology for illegal document reproduction, modification, and distribution.
4.1 Problem Definition

The problem of counterfeiting with scanners and printers has increased substantially in recent years due to the dramatic improvement in scanner and printer capability in terms of resolution and fidelity. These improvements have made the process of counterfeiting printed works better and easier [44]. This has motivated the introduction and development of anti-counterfeiting technology with regard to scanners and printers producing a fake copy.

The list of paper-based valuable documents (see Figure 4.1) extends from high security passports, shares and bonds via medium security admission tickets, gift vouchers etc. to low security purchases, stationery, receipts and bank forms for example [45]. The fact that some types of documents which are considered to be 'low security' does not mean that their abuse is not likely to result in considerable profit for the counterfeiter as well as substantial damage for the rightful owner of such documents. Apart from the security categories of high, medium and low, categories of value may be distinguished in terms of the following [46]:

- **Direct value** documents that have an unconditional and immediate value such as banknotes and gift vouchers.

- **Indirect value** documents that serve to support a transaction or a right, e.g. passports and diplomas.

- **Conditional value** documents that become negotiable only after performing mandatory inspections such as cheques, trading stamps and admission tickets.

- **Informatve value** documents like confidential reports and examination papers.

The security approach depends not only on the security category, but also on the category of value of the document to be protected. Some documents need only to be reproduced to generate considerable profit for the counterfeiter, such as a
4.1 Problem Definition

MEGABANK PLC

Hoard Street Branch, Luton, LU1 0TT

Not to exceed fifty pounds (£50) Date 1st April 2001

Pay Robert Humm & Co £

Mr I Gricer

Cheque No Branch Sort Code Account No Transaction Code
000001 66'6666' 9999999 '02:

Figure 4.1: Printed valuable document.

bank note. Other documents need to be accessed and modified by intruders to get benefits and cause substantial damage for the rightful owner of such documents such as cheques and passports. Consequently three security levels are generally considered:

(i) Copy protection - a strong goal of this level is to prove the originality of the printed work, and to distinguish between original and fake work produced by a scan/print process.

(ii) Ascertaining document integrity (authentication) - ensures that the document has not been altered from the time it was created using a scan/print process.

(iii) Verify the identity of the author - reducing the chance of a forgery.

This thesis concentrates only on the second and the third level (for more information on the first level refer to [47, 48]).

Previous attempts to achieve authentication and document integrity have used traditional watermarking techniques [6]. In these techniques, a watermark
4.1 Problem Definition

(such as the name of the author, his/her signature, or other important information) is embedded into the document in such a way that the contents of the document are not altered. The receiver of the document reconstructs the watermark, which is then used to verify the authors claimed identity or ascertaining the integrity of the document. However, many of the watermarking techniques used are not optimized for printed works and may have difficulties in extracting the watermark after a print-scan attack. This is often due to the extensive amount of noise added to the scanned work.

Other techniques exploit the specific print half-toning process [49]. This scheme has two variants - one for half-tone screens and the other for error diffusion algorithms. The method needs the encoder to have access to the characteristics of the printing process that is not available to any one else.

Some techniques use the principle of fragile watermarking [6]. Fragile watermarks have limited robustness and they are destroyed as soon as the object is modified too much. They are applied to detect modifications of the watermarked data, rather than to convey un-erasable information. Fragile watermarks are destroyed after a scan/print attack; this does not provide a full solution to the problem.

In this thesis, a new approach to digital watermarking for printed documents is developed. This method is robust and it is possible to extract the watermark after a print-scan process. The process is defined by using analytical techniques and concepts used for cryptography. During watermarking, a background is generated using the convolution of the watermark image (e.g. a logo) with an object function using a process termed 'diffusion'. The cover image (text) is inserted into the foreground of the document using a simple additive process termed 'confusion'. The watermark is subsequently recovered by removing the foreground and then correlating the result with the original object function.

This method is explored in detail in the following sections, explaining the
robustness to a wide variety of attacks, including geometric and print/scan attacks. Some applications for the method are also introduced and details of the main degradation associated with printed documents are presented.

4.2 Paper Degradation

Watermarks applied to documents have to withstand a substantial amount of abuse that results in a degradation of the document. These degradations can be divided into two categories: degradations before scanning and degradations during the scanning process. Designing a watermarking scheme to compensate for these kinds of degradation is a challenge [14].

Printed documents do not maintain their quality over time. They are subjected to aging, soiling, crumpling, tearing and deterioration. Moreover, the watermarks have to adapt to different printing process, paper and ink which may degrade the watermark.

After an image is printed and scanned, it can be expected to undergo filtering, rotation (due to alignment errors by the user), scaling, cropping, and have the luminance and contrast adjusted. There are also various sources of noise that may be introduced through, for example, particles of dust in the scanner’s optical system. Moreover, different input devices (scanner, cameras) introduce different types of distortions.

In general, the scanning process follows a customary procedure [50]. First, a user places the original work on a flatbed scanner. Then, the scanner scans the whole area to obtain a low-resolution preview of the work. After this process, the user selects a cropping window to decide on an appropriate range of the work. The scanner then scans the work again with a higher resolution. The final scanned work is obtained by sampling the output of the scanner. Each stage introduces additional degradation into the work, and the counterfeiter will try to minimize
their effects in order to increase the visual similarity between the counterfeit copy and the original. These effects can be summarized as follows [14, 41, 50]:

(i) The scanned version is usually filtered, luminance-contrast adjusted, and gamma corrected.

(ii) The scanned work may be rotated. The angle here is simple, since it is limited to rotation in the scanner's plane (2D). If the work is not well placed, this step may introduce a small orientation error.

(iii) The dimensions of the work may be changed because of cropping and possible scaling. The size of any work captured by a scanner depends on the scanning resolution used. The size of this work is usually different from the original, because the resolution in the scanner and the printer may be different.

(iv) The work may be cropped (i.e., include only a part of the original work, or the whole work with additional background). An associated translation shift might occur.

(v) Some effects are perceptible to the human eye. They affect the visual quality of a scanned work, such as contrast adjustment. Others do not introduce significant effects on the visual quality, but may introduce considerable changes at the signal level, especially on the Fourier coefficients of the scanned work, such as rotation, scaling, and cropping.

4.3 Fundamental Watermarking Model

The image degradation process can be modeled using the 'Fundamental imaging model' for a signal which is given by [51]:

\[ s = \hat{P} f + n. \]
4.3 Fundamental Watermarking Model

In this model a degradation function $p$ (the Point Spread Function or PSF) together with an additive noise field $n$ operates on an input image $f$ to produce the degraded image $s$ [24]. Given $s$, knowledge about the degradation function $p$ and statistical knowledge about the additive noise term $n$, it is possible to obtain an estimate of the original image, $f'$ say. The more information that is known about $p$ and $n$, the closer $f'$ will be to $f$.

This is the basic model that underpins the watermarking technology presented in this thesis. Assuming that $w$ is the information content for the watermark signal (i.e. corresponding to the input image $f$), $\hat{P}$ is some linear operator, $c$ is the cover image (i.e. corresponding to additive noise field $n$) and $w_c$ is the watermarked cover image (i.e. corresponding to the degraded image $s$), then the watermarking model is given by:

$$w_c = \hat{P}w + c$$

In the field of cryptology [51], the operation $\hat{P}f$ (or $\hat{P}w$) is referred to as the processes of 'Diffusion' and the process of adding noise (i.e. $\hat{P}f + n$ is referred to as the process of 'Confusion'. This model is shown diagrammatically in Figure 4.2. It can be shown that if $\hat{P}$ is a linear position-invariant process, then the watermarked cover (degraded image) can be described in the spatial domain by
4.4 Watermark Recovery Model

4.4.1 Private System

Formally, the recovery of $w$ from $w_c$ is based on the inverse process

$$w = \hat{P}^{-1}(w_c - c)$$

where $\hat{P}^{-1}$ is the inverse operator. Clearly, this requires the field $c$ to be known a priori. If this field has been generated using a pseudo-random number generator for example, then the seed used to generate this field must be known a priori in order to recover the data $w$. In this case, the seed represents the private key required to recover $w$. However, in principle, $c$ can be any field that is considered appropriate for confusing the information $\hat{P}w$, including a pre-selected signal or image. Further, if the process of confusion is undertaken in which the signal-to-noise ratio is set to be very low (i.e., $\|c\| >> \|\hat{P}w\|$), then the watermark $w$ can be hidden covertly in the data $c$ provided the inverse process $\hat{P}^{-1}$ is well defined and computationally stable. In this case, it is clear that the host signal or image $c$ must be known in order to recover the watermark $w$, leading to a private watermarking scheme in which the field $c$ represents a key. This field can be lossless compressed and encrypted as required. In addition, the operator $\hat{P}$ (and its inverse $\hat{P}^{-1}$) can be key dependent. The value of this key dependency relies on the nature and properties of the operator that is used, and whether it is compounded in an algorithm that is required to be in the public domain for example.

$$w_c = p \otimes \otimes w + c$$

where $\otimes \otimes$ denotes 2D spatial convolution, and $\hat{P} = p \otimes \otimes$. For the remaining part of this thesis, this model is referred to a 'Fundamental watermarking model'.
4.4 Watermark Recovery Model

4.4.2 Public System

Another approach is to consider the case in which the field $c$ is unknown, and the problem of extracting the watermark $w$ in the absence of this field. In this case, the reconstruction is based on the result:

$$ w = \hat{P}^{-1}w_c + m $$

where

$$ m = -\hat{P}^{-1}c $$

Now, if a process $\hat{P}$ is available in which $\|\hat{P}^{-1}w_c\| > > \|m\|$, then an approximate (noisy) reconstruction of $w$ can be obtained, in which the noise $m$ is determined by the original signal-to-noise ratio of the data $w_c$ and hence, the level of covertness of the diffused watermark $\hat{P}w$. In this case, it may be possible to post-process the reconstruction (de-noising for example) and recover a relatively high-fidelity version of the watermark, i.e.

$$ w \sim \hat{P}^{-1}w_c $$

This approach (if available) does not rely on a private key (assuming $\hat{P}$ is not key dependent). The ability to recover the watermark only requires knowledge of the operator $\hat{P}$ (and its inverse) and post-processing options as required. The problem here is to find an operator that is able to recover the watermark effectively in the presence of the field $c$. Ideally, we require an operator $\hat{P}$ with properties such that $\hat{P}^{-1}c \to 0$.

By using the deconvolution process, we can show that the inverse process is undertaken by correlating the diffused watermark $(w_c - c)$ in a private system (or watermarked cover $w_c$ in a public system) with the (complex) conjugate of the PSF function.
4.5 Deconvolution Process

Deconvolution is a particularly important subject area in signal and image processing. In general, this problem is concerned with the restoration and/or reconstruction of information (e.g. a watermark) from known data (watermarked cover for example). It depends critically on a priori knowledge on the way in which the data has been generated and recorded.

Mathematically, deconvolution is concerned with inverting the process used to form the data. Moreover, there is no exact or unique solution to this problem - it is an ill-posed problem. We attempt to find a 'best estimate' based on a given criterion and specific conditions. In the following sections, three types of filters that can be used in the watermark recovery process are introduced. Those filters are: the Inverse filter, the Wiener filter and the Matched filter.

4.5.1 Inverse Filter

The inverse filter is a straightforward approach to deconvolving the fundamental watermarking equation. In principle, the inverse filter provides an exact solution to the problem when the cover term $c$ can be totally removed. In a private system, where $c$ is a known key, the diffused watermark $\hat{w}$ can be extracted by simply subtracting the cover from the watermarked cover, i.e.

$$\hat{w} = w_c - c = p \otimes \hat{w}$$

The basic approach to solving this equation is to process the data $\hat{w}$ in Fourier space. Using the convolution theorem we have,

$$\hat{W} = PW$$

where $\hat{W}, P$ and $W$ are the (2D) Fourier transforms of $\hat{w}, p$ and $w$ respectively.
Rearranging and taking the inverse DFT, denoted by IDFT, we have

\[ w = \text{IDFT} \left( \frac{\hat{W}}{P} \right) \]

\[ = \text{IDFT} \left( \frac{P^* \hat{W}}{|P|^2} \right) \]

Note that

\[ \frac{1}{P} = \frac{P^*}{|P|^2} \]

which is called the Inverse Filter.

However, in practice, this solution is fraught with difficulties. First, the inverse filter is invariably a singular function. Equally bad, is the fact that even if the inverse filter is not singular, it is usually ill-conditioned. This is where the magnitude of \( P \) goes to zero so quickly, that \( 1/|P|^2 \) rapidly acquires extremely large values. The effect of this is to amplify the (usually) noisy high frequency components of \( \hat{W} \). This can lead to a restoration \( w \) which is dominated by the noise in \( \hat{w} \). The computational problems that arise from implementing the inverse filter can be avoided by using a variety of different filters whose individual properties and characteristics are suited to certain types of data. One of the most commonly used filters for image restoration is the Wiener filter which is considered next.

### 4.5.2 Wiener Filter

Suppose, we were to implement the inverse filter on a digital computer; if \( P \) approached zero (in practice a very small number) for any value of \( P \), then depending on the compiler, the computer would respond with an output such as '... arithmetic fault ... divide by zero'. A simple solution would be to regularize the result, i.e. use

\[ \text{Wiener filter} = \frac{P^*}{|P|^2 + \text{constant}} \]
4.5 Deconvolution Process

so that

\[ w \approx \text{IDFT} \left( \frac{P^* \tilde{W}}{|P|^2 + \text{constant}} \right) \]

and 'play around' with the value of the constant until 'something sensible' is obtained. The constant ideally reflects any available information on the average signal-to-noise ratio of the image. Typically, we consider an expression of the form

\[ \text{constant} = \frac{1}{(\text{SNR})^2} \]

where SNR stands for Signal-to-Noise Ratio. In practice, the user must adjust these parameters until a suitable 'user optimized' reconstruction is obtained. In other words, the Wiener filter must be 'tuned' to give a result which is acceptable based on the judgment and intuition of the user. This interactive approach to image restoration is just one of many practical problems associated with deconvolution which should ideally be executed in real time.

4.5.3 Matched Filter

The Matched Filter is one of the most common filters used for pattern recognition. It is based on correlating an image with a matching template of the feature that is assumed to be present in the image. If the feature does exist, then the output of the filter (the correlation surface) produces a local maximum or spike where the feature occurs.

In watermarking applications, matched filtering is based on correlating the watermarked cover \( w_c \) with the PSF \( p \). The estimate of \( w \) can therefore be written as

\[ w \approx w_c \odot \odot p \]
4.5 Deconvolution Process

Derivation of Matched Filter

Given that

\[ w_c = p \otimes w + c \]

the match filter \( q \) provides an estimate for \( w \) of the form

\[ w \approx q \otimes w_c \]

where \( q \) is chosen in such a way that the ratio

\[ R = \frac{\left| \sum \limits_{\nu} Q P \right|^2}{\sum \limits_{\nu} |C|^2 |Q|^2} \]

is a maximum. The ratio defining \( R \) is a measure of the signal-to-noise ratio. In this sense, the matched filter maximizes the SNR of the output. The matched filter \( Q \) is found by first writing

\[ Q P = |C| Q \times \frac{P}{|C|} \]

and then using the 'Schwarz inequality', i.e.

\[ \left| \sum \limits_{\nu} Q P \right|^2 \leq \sum \limits_{\nu} |Q|^2 \sum \limits_{\nu} |P|^2 \]

giving

\[ \left| \sum \limits_{\nu} Q P \right|^2 = \left| \sum \limits_{\nu} |C| Q \frac{P}{|C|} \right|^2 \]

\[ \leq \sum \limits_{\nu} |C|^2 |Q|^2 \sum \limits_{\nu} \left| \frac{P}{|C|} \right|^2 \]

From this result, and the definition of \( R \) given above we get

\[ R \leq \sum \limits_{\nu} \left| \frac{P}{|C|} \right|^2 \]
4.5 Deconvolution Process

Now, recalling that the criterion for the matched filter is that $R$ is a maximum, if this is the case then,

$$R = \sum_{i} \frac{|P|^2}{|C|^2}$$

or

$$\left| \sum_{i} |C| Q \frac{P}{|C|} \right|^2 = \sum_{i} |C|^2 |Q|^2 \sum_{i} \frac{|P|^2}{|C|^2}$$

This is true, if and only if

$$|C| Q = \frac{P^*}{|C|}$$

Thus, $R$ is a maximum when

$$\text{Matched Filter} = Q = \frac{P^*}{|C|^2}$$

Matched Filter for a White Cover

If the cover $c$ is white noise, then its power spectrum is uniformly distributed. In particular, under the condition

$$|C|^2 = C_0$$

we have

$$Q = \frac{P^*}{C_0}$$

and

$$W \approx \frac{P^* W}{C_0}$$

According to Fourier transform property [24, 35], multiplying the Fourier domain by a constant is equivalent to multiplying the spatial domain by the same constant. Moreover, because the output of the inverse Fourier transform will be normalized between zero and one, this means that the constant $c_0$ will be canceled automatically. Hence, for white noise, the match filter provides an estimate which may be written in the form

$$w \approx w_c \odot \odot p$$
4.5 Deconvolution Process

Matched Filter for a Private System

In a private watermarking system, the cover \( c \) is known and can be removed before applying the filter. In this case, the diffused watermark can be represented by

\[
\hat{w} = w_c - c = p \otimes w
\]

\[
= p \otimes w + c_1
\]

where \( c_1 \) is a black image (i.e. zero image). The power spectrum \( |C_1|^2 \) for each component in the frequency space for this image is equal to zero. In order to avoid division by zero in the matched filter, a similar technique is used to that already shown in weiner filter i.e.

\[
W \approx \frac{P^* \hat{W}}{|C_1|^2 + \text{constant}}
\]

This constant will be canceled in the spatial domain (see above), therefore the solution for \( w \) is given by

\[
w \approx \text{IDFT}[P^* \hat{W}]
\]

or

\[
w \approx \hat{w} \otimes \circ p
\]

\[
\approx p \otimes w \otimes \circ p
\]

which in Fourier space is

\[
W \approx P W P^* = |P|^2 W
\]

or

\[
w \approx \text{IDFT} \left( |P|^2 W \right) \tag{4.1}
\]

Observe, that the amplitude spectrum of the reconstructed watermark is given by \( |P|^2 |W| \) and the phase information is determined by \( W \) alone.
4.5.4 Adaptive (Combination) Filtering

According to Equation (4.1), the matched filter reconstructs a noisy version of the embedded watermark. This noise is due to the power spectrum of the PSF (i.e. $|P|^2$). In order to enhance the extracted watermark, $|P|^2$ needs to be eliminated or at least, minimize its effect. An algorithm is therefore proposed for adaptive filtering that aims to enhance the extracted watermark. This algorithm is not applicable unless there is a full control over the PSF ($p$) used in the diffusion process. There are some options that can be considered in order to eliminate the power spectrum of the PSF:

(i) Divide over the power spectrum $|P|^2$ during the diffusion step. Then, use the matched filter to extract an exact version of the watermark. The diffused watermark becomes

$$\hat{w} = \text{IDFT} \left( \frac{PW}{|P|^2} \right)$$

and the extracted watermark is

$$w = \text{IDFT} \left( \frac{PWP^*}{|P|^2} \right) = \text{IDFT} (W)$$

To avoid singularities, each zero in $|P|^2$ is replaced by 1. This will not change the result since the quantity $WP^*$ is zero, but will effectively remove the 'division by zero' problem.

(ii) Pre-process the diffusion operator $p$ - before the diffusion step - by replacing the amplitude spectrum of the PSF with a constant value (i.e. $|P| = \text{constant}$). Multiplying the Fourier domain by a constant is equivalent to multiplying the spatial domain by the same constant. This constant in the spatial domain has no effect on the image (since the image will be normalized between zero and one). Consequently, there is no need to divide over
4.5 Deconvolution Process

the power spectrum in the Fourier domain. The reconstructed watermark is reduced to: \( w = \text{IDFT}(W) \). This gives an exact result.

(iii) Choose a PSF that has a homogeneously distributed power spectrum \( |P|^2 \) across all frequencies (i.e. the spectrum is more or less flat). For example, use a white noise as a diffusion operator. This implies that the changes of the power spectrum between frequencies are too small. Therefore, we can assume - to a good approximation - that the power spectrum is constant over all frequencies. The watermark can be extracted using the matched filter with minimal distortion. It is possible to post-process the reconstruction to improve the fidelity of the recovered watermark (such as thresholding). On the other hand, if we divide through by the power spectrum, then we obtain an exact reconstruction.

(iv) If the power spectrum is not uniformly distributed, then the matched filter (i.e. ignoring the division by the power spectrum of \( p \)) will not reconstruct an acceptable version of the watermark. In such cases, the inverse filter is used instead of the matched filter. Note, that the power spectrum may be changed dramatically and may approach zero quickly, producing the ill-conditioned problem already described. In this case, by dividing through by the power spectrum, it is possible to extract a very distorted (dominated by noise) version of the watermark. To overcome this problem, the following compromise can be made: first, extract the watermark without division over the power spectrum. If the extracted watermark is clear enough then there is no need to undertake another process. Otherwise, start the process of enhancing the result by dividing the frequency domain by the power spectrum, starting from frequencies with high power spectral values and then progress toward the lowest values until an acceptable extraction is obtained. In this way, an acceptable extraction can be achieved before the high frequency component values of the power spectrum are introduced (i.e.}
those responsible for ‘driving’ ill-conditioned behaviour).

(v) Use the Wiener filter instead of the inverse filter in the previous option. The user must adjust the parameters of this filter until a suitable ‘user optimized’ reconstruction is obtained.

Division over the power spectrum during the diffusion step (option 1) is preferable to that of the extraction process (options 4 and 5). This is because the watermarked cover is vulnerable to translation and/or rotation during the scanning process and the frequency components of the extracted watermark will be divided by values that are not related to it. This requires the two images to be registered (image registration is the process of transforming one image into the coordinate system of another image [52]) before applying the division. On the other hand, division during the diffusion step has no matching problem.

In summary, an acceptable version of the watermark can be extracted by correlating the diffused version with the PSF. If this does not produce acceptable result for any particular reason, the result of the matched filter can be enhanced by applying options 1, 2, or 3 during watermarking, or options 4 or 5 during the watermark extraction process.

4.6 Embedding and Extraction Algorithms

In the previous section, a mathematical model for the embedding and extraction process was discussed. In this section, an implementation of this model - for paper document security - is presented. The implementation is described using pseudo-code that provides a self-description. Additional comments or explanations are also presented between procedures or after the comment symbol (%).
4.6 Embedding and Extraction Algorithms

4.6.1 Embedding Algorithms

In order to protect a text document, a watermark (e.g. logo) can be embedded into the background of the document; then, the text is placed in the foreground. The watermark is diffused using a diffusion operation. The method is described by the following algorithms and illustrated in Figure 4.3.

Algorithm 4.1: Embedding a watermark

```plaintext
[wc, key] = Embed() {
    w = LoadWatermark(); % black/white or grey watermark
    c = LoadCover(); % This is the text document
    [cols, rows] = size(c);
    while the user is not satisfied do
        [specific-keys, diffusion-type, p] =
            GenerateDiffusedOperator(cols, rows);
        % select preprocessing options
        preprocessing = SelectPreProcessing();
        if preprocessing = 1 % option 1
```
The function $Embed$ simply calls the corresponding sub-procedures related to the watermarking process. It starts by loading the watermark and the cover images. The diffused operator is then generated depending on the user selection. If preprocessing is required, then the code executes the selected process. Once the diffusion operator is ready, the diffusion procedure is called to produce the diffused watermark. Finally the cover is confused with the diffused watermark. The output of this procedure is the watermarked cover and the key. The key is the combination of all data needed in the extraction process, namely: the diffusion operator type; specific parameters that are necessary to regenerate the operator in the extraction process; application of the pre-processing applied to the diffusion operator; and size of the cover image. The user can examine the output (watermarked cover). If he/she is satisfied with the output, then it can be printed, otherwise, the diffusion
4.6 Embedding and Extraction Algorithms

operator can be changed and the process repeated.

Algorithm 4.1.1: Generation of the diffusion operator

```plaintext
[specific-keys, type, p] = GenerateDiffusedOperator(cols, rows) {
    type = ChooseTheDiffusionType();
    if type = 'Fresnel' then
        specific-keys = read the related parameters
        to Fresnel function
    elseif type = 'Gaussian' then
        specific-keys = read the related parameters
        to Gaussian function
    elseif type = 'other'
        ...
    endif
    p = GenerateTheOperator(specific-keys, type, cols, rows)
}
```

The GenerateDiffusedOperator lists some operators that can be used in the diffusion process (see Figure 4.4). The user selects the operator and enters the related parameters (see Figure 4.5). Finally, the diffusion operator image is generated to cover the whole background of the text document. The output of this procedure is the operator itself, its type and specific parameters.

Algorithm 4.1.2 Divide through by the power spectrum

```plaintext
p = DivideOverThePowerSpectrum(p){
    P = DFT(p);
    For each frequency component, \( P_{ij} \), do
    \[
    PS_{ij} = \text{Re}(P_{ij})^2 + \text{Im}(P_{ij})^2; \quad \% \text{Im: Imaginary part}
    \]
```
4.6 Embedding and Extraction Algorithms

The *DivideOverThePowerSpectrum* implements option 1 from the adaptive filtering processes. It starts by computing the Fourier transform of the input diffused operator $p$. The power spectrum is then computed. In order to avoid 'division by zero', all zeros must be replaced by one. The output of this procedure is the diffused operator after having been divided by the power spectrum. Figure 4.6 shows the interface that is used during the diffusion process. Two buttons (*with pre-processing* and *without pre-processing*) are available to the user in order to select the method required.
4.6 Embedding and Extraction Algorithms

**Algorithm 4.1.3** Regularize the power spectrum

```plaintext
p = RegularizePowerSpectrum(p) {
    P = DFT(p);
    For each frequency component P_{ij} do
        \theta_{ij} = \tan^{-1}(\text{Im}(P_{ij})/\text{Re}(P_{ij}))
    endFor
    p = IDFT(e^{i\theta}); % i=\sqrt{-1}
}
```

The `RegularizePowerSpectrum` implements option 2 from the adaptive fil-

**Figure 4.5:** Generating the diffusion operator interface.
Figure 4.6: Generating diffused watermark interface.
tering menu First, the procedure transforms $p$ to the Fourier domain. It then extracts the phase components only (i.e. neglects the amplitude spectrum). This is equivalent to setting the power spectrum to one. It then returns back to the spatial domain. The output is the modified $p$ (pre-processed). The box that must be checked to apply this process is shown in Figure 4.5 (CLEAR AMPLITUDE).

Algorithm 4.1.4: Diffuse the watermark with the diffusion operator

```plaintext
pw = Diffuse(p, w) {
    [cols, rows] = size(p);
    w = extend (w, cols, rows);
    P = DFT(p);
    W = DFT(w);
    PW = P*W; % Dot Product
    pw = IDFT (PW);
    pw = real (pw);
    pw = Normalize (pw);
}
```

In the Diffuse procedure, the watermark is diffused with $p$ using a second-order convolution in the Fourier domain. According to the ‘convolution theorem’, the convolution of two images in real space is the same as the product of their Fourier transforms in Fourier space [51]. Typically, the watermark image is small compared to the document size. Thus, the watermark should be zero-padded until its size becomes identical to the document size. After reverting back to the spatial domain, the imaginary part - if any - must be discarded. Finally, the output is normalized, giving a floating point output between 0.0 (black) and 1.0 (white).
Algorithm 4.1.5: Confuse the watermark with the cover.

```plaintext
1 wc = Confuse(pw, c, α){
2   wc = c + α * pw;
3   wc = Normalize(wc);
4 }
```

for Confusion, the diffused watermark, \( pw \), is added to the cover image \( c \) using linear addition. The linear addition is represented by \( wc = c + αpw \), where \( α \) is the embedding strength. The output is then normalized.

### 4.6.2 Extraction Algorithms

The recovery process involves removing the foreground and then correlating the result with the same diffusion operator used in the embedding process. This method is described by the following algorithm and illustrated in Figure 4.7.

![Figure 4.7: Extraction process.](image-url)
Algorithm 4.2: Extract the watermark

```plaintext
w = Extract(wc, key, c) {
    Parse the key to its main components:
    [cols, rows, diffusion-type, preprocessing-type and specific-keys];
    p = GenerateTheOperator(specific-keys, diffusion-type, cols, rows)
    if preprocessing-type = 2 % option 2
        p = RegularizePowerSpectrum(p);
    endif
    wc = scale(wc, rows, cols);
    if c != NULL
        wp = RemoveCover(wc,c);
    endif
    w = De-diffuse(wp, p);
    while the user can not recognize the watermark
        input the filter type into 'option'; % inverse or wiener filter
        w = postprocess(w,p,option);
        if the user can recognize the watermark
            status= 'watermarked'
            Exit while
        elseif the user stop refinements
            status= 'un-watermarked'
            Exit while
        endif
    endwhile
}
```

The inputs of the Extract procedure are as follows: the watermarked document wc; the key used in the extraction process; the cover - if available. The input
document may be the result of a scan or an *a priori* computer file. The procedure starts by breaking the key into its main components:

- **Cols and rows** - the size of the original document.
- **Diffusion-type** - the type of the diffusion operator used in the diffusion process.
- **Preprocessing-type** - division by the power spectrum or amplitude regularization
- **Specific-keys** - the values of all the parameters used to generate the diffusion operator.

The procedure then generates the diffusion operator used by the embedding process applying the same parameters. If the operator has been regularized before, then it needs to be regularized again. The watermarked cover must be scaled to its original size (e.g. if the input document is the output of a scanner). To enhance the result of this procedure the cover must be removed, leaving the diffused watermark only. Correlation is then applied to the diffused watermark with the diffusion operator. Finally, post-processing can be applied, e.g. developing a partial inverse filter - until an acceptable output is obtained.

**Algorithm 4.2.1: Remove the cover**


```
1 wp = RemoveCover(wc,c){
2     wp = wc - SNR*c;
3     wp = Normalize(wp);
4 }
```

The *RemoveCover* procedure is based on subtracting the watermarked cover from the known cover. This way is applicable in a private system where the cover is
given as a known key. The two images \((w_c\text{ and } c)\) should have the same coordinate system (in terms of registration). Otherwise, one image will be subtracted from a de-registered (shifted) version of the other giving poor results. On the other hand, in public system - where the cover is not known - another way to remove the cover must be sought.

**Algorithm 4.2.2: De-defuse the watermark**

```
1 w = De-diffuse(wp,p){
2 P = DFT(p);
3 W = DFT(wp);
4 W = conj(P)*W; % conj: Conjugate of P
5 w = IDFT (W);
6 w = real (w);
7 w = Normalize (w);
8 }
```

In the *De-diffuse* procedure, the watermark is extracted - if any - using correlation in the Fourier domain. As in convolution, correlation has an equivalent representation in Fourier space namely, if

\[
\hat{s} = \hat{p} \circ \hat{f}
\]

then

\[
\hat{S} = \hat{P}^* \hat{F}
\]

where \(P^*\) is the complex conjugate of \(P\). This result is known as the correlation theorem [51]. After going back to the spatial domain, the imaginary component - if any - is discarded and the output normalized.
Algorithm 4.2.3: Post processing (refinements)

```plaintext
1 w = Postprocess(w, p, option){
2     input a threshold into thr
3     W = DFT(w);
4     P = DFT(p);
5     For each frequency component, P\textsubscript{ij} do
6         PS\textsubscript{ij} = Re(P\textsubscript{ij})\textsuperscript{2} + Im(P\textsubscript{ij})\textsuperscript{2};
7         if PS\textsubscript{ij} >= thr
8             if option='inverse'
9                 if PS\textsubscript{ij} = zero then PS\textsubscript{ij} = 1;
10                 W\textsubscript{ij} = W\textsubscript{ij} / PS\textsubscript{ij} ;
11             elseif option = 'wiener'
12                 W\textsubscript{ij} = W\textsubscript{ij} / (PS\textsubscript{ij} + constant) ;
13             endif
14         endif
15     endFor
16     w = IDFT(W);
17     w = real(w);
18     w = Normalize(w);
19 }
```

This procedure implements the ideas mentioned in option 4 and 5 for adaptive filtering. A threshold is specified and all frequency components $W\textsubscript{ij}$ that correspond to frequency components $P\textsubscript{ij}$ with a power spectrum greater than the threshold are used to compute the inverse or Wiener filter. Figure 4.8 shows the interface used to implement this procedure. Two sliders - one for the inverse filter and the other for the Wiener filter - are used to change the threshold.
4.7 Fresnel Coding

The Fresnel transform is the basis for Fresnel optics, which is derived by studying the properties of optical diffraction (see Section 3.5.2). The Fresnel transform describes the wave propagation in the Fresnel diffraction region and can be derived...
4.7 Fresnel Coding

using the Kirchhoff theory of diffraction as shown in Appendix A.1. Here, it is shown that the Fresnel transform can be represented by the following formula:

\[ f_{xy} = \exp(\alpha x^2) \exp(\imath \alpha y^2) = \exp(\imath \alpha (x^2 + y^2)) \]

where \( \alpha \) is a Fresnel constant.

The Fresnel transform can be used to diffuse a watermark (see Figure 4.10(a)). An advantage of using Fresnel rings as a diffusion operator is that the image has a spread spectrum in the Fourier domain and a central symmetry so that the spectrum is uniform in all directions. The choice of ring density (in other words, the number of visible rings per image - see Figure 4.9) is crucial (controlled by \( \alpha \)). Low density rings (with minimal high frequency content) yields blurring of the recovered watermark; high density rings result in a recognizable extracted watermark.

![Fresnel rings: (a) Thin rings(\( \alpha = 0.01 \)), (b) Thick rings(\( \alpha = 0.003 \))](image)

**Figure 4.9:** Fresnel rings: (a) Thin rings(\( \alpha = 0.01 \)), (b) Thick rings(\( \alpha = 0.003 \))

4.7.1 Fresnel Results (e-version)

**Grey-scale fresnel coding**

In Figure 4.10, the Fresnel function has been used to diffuse the watermark. Matched filtering is then used to extract the watermark. The parameters for
4.7 Fresnel Coding

Figure 4.10: Grey-scale fresnel coding: (a) Diffused watermark, (b) Watermarked cover, (c) The watermark, (d) Extracted watermark after the cover image has been subtracted, (e) Extracted watermark if cover is not subtracted; Note that the watermark extraction was unsuccessful if the cover is not subtracted.

This experiment are as follows: the Fresnel constant is 0.01; cover image size is $200 \times 200$, watermark image size is $86 \times 55$. The watermark is the 'Communication and Signal Processing Division' logo in Loughborough University (Figure 4.10 (c)), the embedding strength is 10%, subtraction SNR is 100% and both the cover and the watermark are grey scale images. Figure 4.10 (a) shows the diffused watermark. Figure 4.10 (b) shows the cover image after it has been watermarked. Figure 4.10 (d) shows the extracted watermark after the cover has been subtracted. The experiment also shows that the watermark can not be extracted without subtracting of the cover. The extracted watermark is clear enough for post-processing not to be required, the method being applied to an electronic version of the watermarked cover (i.e. computer file).
4.7 Fresnel Coding

Document Fresnel Coding

This investigation is the same as that described above, except that the cover is a valuable document image - a named cheque (Figure 4.1). The parameters for this experiment are as follows: Fresnel constant is 0.008; cover image size is 247 x 587; watermark image size is 64 x 64. The watermark is the author's signature (Figure 4.11 (b)), embedding strength is 100% and subtraction SNR is 55%. The results are shown in Figure 4.11. Figure 4.11 (a) shows the cover image after it has been watermarked. Figure 4.11 (c) shows the extracted watermark after applying the matched filter directly on the watermarked cover (i.e. without removing the foreground). The extracted watermark after removing the foreground (private system) is shown in Figure 4.11 (d). The results demonstrate that removing the foreground increases the fidelity of the extracted watermark.

In document watermarking, the diffused watermark can be placed in the background of the document using a strong embedding strength with minimal effect on the content. On the other hand, increasing the embedding strength in image (not document) watermarking decreases the fidelity of the cover image. Therefore, the watermarked document has greater robustness than the watermarked image.

Adaptive Filtering: Option 1

This option provides a technique whereby the embedding data is divided by the power spectrum of the diffusion operator. Figure 4.12(a) shows the resultant watermarked cover. The result of applying the matched filter on the diffused watermark directly (i.e. before adding the cover) is shown in Figure 4.12(b) after applying a threshold (threshold=60%) on the extracted watermark. Note, that if the watermark is extracted after adding the cover, a null result is obtained, even if the cover is subtracted later.

Comparing this result with the above experiment (i.e. without any pre-
4.7 Fresnel Coding

Figure 4.11: Document fresnel coding. (a) Watermarked cover, (b) The watermark, (c) Extracted watermark without removing the cover, (d) Extracted watermark after removing the cover.
Figure 4.12: Adaptive filtering (option 1) (a) The watermarked document, (b) Extracted watermark without adding the cover, (c) Extracted watermark after adding the cover (unsuccessful), (d) Extracted watermark after subtracting the cover (unsuccessful).
processing) reveals that 'division by power spectrum' decreases the robustness of the watermarked cover. Consequently, this option is not suitable for document watermarking. On the other hand, the watermark can be extracted if the cover is not added (i.e., not used) to the diffused watermark.

**Adaptive Filtering: Option 2**

This regularize the amplitude of the diffusion operator before diffusion. Figure 4.13(a) shows the resultant watermarked cover. The result of applying the matched filter on the diffused watermark directly (i.e., before adding the cover), after adding the cover, and after adding and then subtracting the cover are shown in Figure 4.12(b, c and d) respectively.

Comparing the result of this experiment with the extracted watermarks in Figure 4.11 demonstrates an improvement on previous results.

**Adaptive Filtering: Option 3**

This option uses white noise as a diffusion operator. Since the Fresnel function is not white noise, this option is examined in the following chapter.

**Adaptive Filtering: Options 4 and 5**

These options use the inverse or Wiener filters in the extraction process. This experiment shows that both filters can produce perfect outputs before saving the watermarked cover (i.e., without any noise). Figures 4.14 (b and f) show the extracted watermark after applying inverse and Wiener filters respectively. Neither filters extracts any feasible output after saving and loading the images or adding the foreground.
Figure 4.13: Adaptive filtering (option 2): (a) The watermarked document, (b) Extracted watermark without adding the cover, (c) Extracted watermark after adding the cover, (d) Extracted watermark after subtracting the cover.
4.7 Fresnel Coding

Figure 4.14: Adaptive filtering (options 4 and 5): (a) The watermarked document, (b, c, d and e) Extracted watermark after applying the inverse filter, (f, g, h and i) Extracted watermark after applying the Wiener filter. Note that unsuccessful extraction is presented by a cross image.
4.7.2 Fresnel Results (scanned-version)

Testing this model after scanning using low-resolution and after rescaling the scanned image to its original size does not provide for extraction of the watermark.

4.8 Summary

(i) Using division (options 1, 4, and 5) decreases the robustness and makes these options useless especially for document or image watermarking. They might be used with diffused watermark only (i.e. without adding any cover).

(ii) In order to make division useful, the two images must be registered first. Any spatial shift - even of one pixel - yield a poor output.

(iii) Options 2 and 3 provide enhancement to the result.

(iv) We do not have to use these options unless the matched filter fails to produce a satisfactory output.

(v) Since the Fresnel function cannot resist a print/scan attack, we have to look for another diffusion operator.

In the following chapter, we investigate the use of different types of noises as a diffusion operator and establish whether we can extract the watermark after a print/scan attack.
Chapter 5

Document Noise Coding

In the previous chapter, the underlying principle of a novel method of watermarking was described. The convolution operation was used along with the Fresnel function to achieve watermarking. Theoretically, any other broad-band signal can provide a basis for an invertible transformation, e.g., a series produced by a chaotic system or a pseudo-random generator can be used instead of Fresnel rings. In this chapter, different types of noises are tested as diffusion operators. The extracted watermarks from electronic and scanned documents are presented. Moreover, the model is extended to include full colour documents. Finally, the main operating features that make this system work are listed.

5.1 Gaussian Noise Coding

In this section, Gaussian noise is used as a diffusion operator. A brief description of the Gaussian model is described below. In order to enhance the coding and the extraction algorithm, a modification of the algorithms that were introduced in the previous chapter are presented called Confusion and RemoveCover. The experiments were applied to a document image of size 461 \times 356 and a 256 \times 128 watermark as shown in Figure 5.1 and 5.2 respectively.
5.1 Gaussian Noise Coding

Figure 5.1: Original document used for Gaussian coding.
5.1 Gaussian Noise Coding

5.1.1 Gaussian Noise

This model (also called normal) is frequently used in practice because of its mathematical tractability in both the spatial and frequency domain [24]. The probability distribution function (PDF) is given by

\[ p(z) = \frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \]

where \( \mu \) is the mean of average values of \( z \), and \( \sigma \) is the standard deviation. The standard deviation squared, \( \sigma^2 \), is called the variance of \( z \). Approximately 70% of the \( z \) values will be in the range \( (\mu - \sigma), (\mu + \sigma) \) and about 95% will be in the range \( (\mu - 2\sigma), (\mu + 2\sigma) \). A plot of this function is shown in Figure 5.3.

5.1.2 Modified Confusion Algorithm

The confusion process is a simple additive process. The background is added to the foreground (text) using linear addition. This reduces the clarity of the text. To illustrate this, suppose that a portion of the text is written using a specific colour. Since the background is a noisy texture, text pixels will be added to different colour values from the background. Consequently, the text’s colour will be changed from one colour to a multi-colour. In order to enhance the visibility of the text, the following modified confusion algorithm is proposed. The idea of this algorithm is
5.1 Gaussian Noise Coding

Figure 5.3: Gaussian noise: (a) Texture, (b) Histogram, (c) Power spectrum.
Based on a simulation of the printing process, where the printer is 'fed' with plain paper that already has the background printed on it. The printer prints on this paper without mixing the background with the foreground.

Algorithm 5.1: Modified confusing algorithm.

```plaintext
1 \(wc = \text{ModConfuse}(pw, c, \alpha , \text{thr})\) {
2 \(pw = pw \cdot (c \geq \text{thr})\);
3 \(wc = c + \alpha \cdot pw\);
4 \(wc = \text{Normalize}(wc)\);
5 }
```

In `ModConfusion`, the background is divided into two parts according to a threshold \(\text{thr}\); all texture (background) pixels that correspond to foreground pixels with an intensity value less than or equal to \(\text{thr}\) are not used in confusing process. Since the background in the non-watermarked document is white, then a threshold value greater than or equal 0.9 can be used. Now, the linear addition will ensure that the text pixels will be added to zeros values (i.e. will not be changed) and normalized in the same way.

### 5.1.3 Embedding Results

Using Gaussian noise as a diffusion operator is equivalent to applying ‘Option 3’ from the adaptive filtering class. The power spectrum of Gaussian noise (with a seed=10) (see Figure 5.3 (c)) is more or less flat. The watermark can be extracted using the matched filter with minimal distortion. The resultant watermarked document is shown in Figure 5.4. This output is executed without applying any preprocessing. The embedding strength used here is 1 (\(\alpha = 1\)). Figure 5.5 and 5.6 show the resultant watermarked document after applying preprocessing, option 1 and option 2 respectively from the adaptive filtering. Note, that as a result of using the modified algorithm for confusion, the text is not mixed with the background.
5.1 Gaussian Noise Coding

Figure 5.4: Gaussian coding: option 3 (without pre-processing).
Figure 5.5: Gaussian coding, option 1.
Figure 5.6: Gaussian coding: option 2.
5.1.4 Foreground Removal

In order to increase the fidelity of the extracted watermark, it is necessary to eliminate the effect of adding text. This can be accomplished by either removing the added text or forcing the correlation between the added text and the diffused watermark to zero. If a copy of a non-watermarked text is available for extraction, then a text subtraction can be applied. In most systems, the non-watermarked text is not available, and so another technique is needed to remove the text. One technique is described below.

Since the background of the non-watermarked document is white, adding this image to the diffused watermark will increase the intensity of all texture pixels, whilst keep all the text pixels unchanged. This is clear from the histogram analysis (see Figures 5.7 (a) and (b)). This means that the text’s pixels will create impulses in the spatial spectrum of the output image. Those impulses can be removed using a ‘Median Filter’, which is particularly effective in the presence of impulse noise. This method is described in detail by the following algorithm.

**Algorithm 5.2. Eliminating the foreground using a modified median filter**

```plaintext
1  wp = RemoveCover(wc);
2      wsize = input the window size related to Median Filter.
3      T = input a threshold.
4    For each pixels, p_ij in wc do
5        if the intensity of p_ij is less than T then
6            n = Find the neighborhood of p_ij according to wsize.
7            n = Excluding from n all points less than the T.
8            med = Find the median of this window n.
9            p_ij = med.
10       endif
11    endfor
12  wp = Normalize(wc);}
```
5.1 Gaussian Noise Coding

Figure 5.7: Histogram analysis for 'removing foreground' algorithm: (a) Histogram before adding text, (b) Histogram after adding text, (c) Histogram after removing text. Note: y-axis represents the number of pixels for each grey level.
5.1 Gaussian Noise Coding

Here, the algorithm starts by specifying a suitable value for the window size; a large window size is preferred (e.g., 11 x 11). The threshold $T$ can be specified by checking the histogram of the image. A suitable value for $T$ could be the central grey level (i.e., $T = 0.5$). Figure 5.8 shows the interface used to specify the parameters related to this algorithm. All pixels with intensity less than $T$ are likely to be those related to the foreground (i.e., text). Consequently, the median filter is applied to those pixels, excluding all points less than the threshold $T$ from the neighborhood (This ensures that the result of the median is not an impulse also.) Figure 5.7 (c) shows the histogram of the watermarked document after removing the text.

5.1.5 Extracted Watermark Results

Electronic Document

In this section, the system is tested on electronic documents (i.e., without printing and scanning). The experiments cover all possible cases, which depend on the following factors:

(i) The embedding method.

(a) Normal embedding - this is based on the application of option 3 from the adaptive filtering because the Gaussian noise is a white noise Fig-
5.1 Gaussian Noise Coding

ure 5.4 shows the output document.

(b) Embedding using option 1 - Figure 5.5 shows the output document

(c) Embedding using option 2 - Figure 5.6 shows the output document.

(ii) The existence of the text - extract the watermark

(a) Before adding text.

(b) After adding text.

(c) After adding text and then removing it

(iii) Post-Processing - whether option 4 and 5 had been applied or not.

The extracted watermarks for all cases are presented in Table 5.1. The results show that:

(i) Using white noise for diffusion helps the extraction process to extract an acceptable output.

(ii) Extracting the watermark before removing the text produces a very distorted output.

(iii) The 'Foreground Removing' algorithm proves its ability to enhance the results.

(iv) Applying options 1 or 2 is helpful to extract a good output (especially for diffusion-only watermarks).

(v) Applying post-processing enhance the results. However, post-processing is not robust for many attacks, especially geometric attacks.
### Table 5.1: Extracted watermarks after Gaussian coding

<table>
<thead>
<tr>
<th>Option</th>
<th>No Text</th>
<th>Without Subtract Cover</th>
<th>After Subtract Cover</th>
<th>Post Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td>Option1</td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Option2</td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
<td><img src="image9.png" alt="Image" /></td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>
5.1 Gaussian Noise Coding

Scanned Document

Experiments are applied on scanned documents. All documents were printed using an HP Laser Jet 1300 and scanned using an EPSON Perfection 1660 Photo scanner. The scanner resolution was 72 dpi with B/W image type. The extracted watermarks for all cases are presented in Table 5.2. The results show that:

(i) The system succeeds in extracting the watermark from the scanned document using low resolution scanning

(ii) There is no clear differences between the available options. However, option 2 produces better outputs.

Damaged Document

Printed documents do not maintain their quality over time. They are subject to aging, soiling, crumpling, tearing, and deterioration. Figure 5.9, shows a document that has been crumpled, teared and 'scribbled' on. Figure 5.10, shows an acceptable watermark, even though the added noise was extremely large

Extraction using Different Diffusion Field

In this experiment, different diffusion operators - other than one used in coding - were used in the extraction process, for example, using a different seed (key) to generate the noise. The results are given in Figure 5.11 which shows that the system is very sensitive to the key. Only an exact match recovers the watermark; all other combinations generate a noise-like image.
### Table 5.2: Extracted watermarks after Gaussian coding: scanned-version

<table>
<thead>
<tr>
<th>No Text</th>
<th>Without Subtract Cover</th>
<th>After Subtract Cover</th>
<th>Post Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
<tr>
<td>Option1</td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td>Option2</td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
<td><img src="image9.png" alt="Image" /></td>
</tr>
</tbody>
</table>

- Not Applicable
5.1 Gaussian Noise Coding

Figure 5.9: Damaged document.

The Royal Environmental

Advanced Food Hygiene

PASS

This is to certify that

Nick James

has attended the course in Advanced Food Hygiene
organized by the Institute, and, after Examination,
has been awarded this Diploma.

this third day of October 1997

President of the Institute

Course Organizer

Certificate No. 31096

Venue: Amman
5.1 Gaussian Noise Coding

Figure 5.10: Extracted watermark from damaged document.

Figure 5.11: Extracted watermark using Gaussian noise (seed=35).
5.2 Different Noise Types

In the previous section, the Gaussian noise was used as a diffusion operator. In this section, different types of noises are used as a diffusion operator. All experiments were performed on the gift voucher and the watermark which is shown in Figure 5.12 (a) and (b) respectively. For each noise, the extracted watermarks for electronic and scanned documents are given. The purpose of these experiments is to prove/disapprove the suitability of different noise types in this system. Using different types of noises in this system helps to:

(i) Increase the flexibility and diversity of the background, the background texture depending on both the watermark and the noise field used.

(ii) Increase the security of the system; the attackers need to know the type of noise (i.e., the key) in addition to the seed used for generating the noise field.

Noise can be characterized by a probability density function (PDF). The following are some common PDFs found in image processing [24] (for a random
5.2 Different Noise Types

variable $x$, the probability $p(x = X)$ is frequently denoted by $f(x)$; this function is called PDF).

5.2.1 Uniform Noise

The PDF of Uniform noise is given by

$$p(z) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$$

The mean and variance of this density are given by

$$\mu = \frac{a + b}{2}$$

$$\sigma^2 = \frac{(b-a)^2}{12}$$

A 2D plot of the of this function and its histogram are shown in Figure 5.13 (a) and (b) respectively. The power spectrum of this noise is flat (like Gaussian noise). The resultant watermarked voucher is shown in Figure 5.14 and the extracted watermarks are shown in Table 5.3. The results show that uniform noise can be used in this system.
5.2 Different Noise Types

Figure 5.14: Uniform coding: without pre-processing.

Table 5.3: Extracted watermarks after uniform coding.
5.2 Different Noise Types

5.2.2 Gamma (Erlang) Noise

The PDF of Erlang noise is given by
\[ p(z) = \begin{cases} 
\frac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq 0 \\
0 & \text{for } z < 0 
\end{cases} \]

where the parameters are such that \( a > 0 \), \( b \) is a positive integer, and \( ! \) denotes the factorial. The mean and variance of this density function are given by
\[ \mu = \frac{b}{a} \]
and
\[ \sigma^2 = \frac{b}{a^2} \]

A plot of the of this function and its histogram are shown in Figure 5.15 (a) and (b) respectively. The power spectrum of this noise is flat (like Gaussian noise). The resultant watermarked voucher is shown in Figure 5.16, and the extracted watermarks are shown in Table 5.4. The results show that Gamma noise can be used in this system.

Figure 5.15: Gamma noise (a) Texture, (b) Histogram.
5.2 Different Noise Types

Figure 5.16: Gamma coding, without pre-processing.

Table 5.4: Extracted watermarks after Gamma coding.
5.2 Different Noise Types

5.2.3 Poisson Noise

The PDF of Poisson noise is given by

\[ p(z) = \frac{e^{-z}z^a}{a!} \]

where \( a \) can be any non-negative integer. A plot of the function and its histogram are shown in Figure 5.17 (a) and (b) respectively. The power spectrum of this noise is flat (like Gaussian noise).

The resultant watermarked voucher is shown in Figure 5.18, and the extracted watermarks are shown in Table 5.5. The results show that Poisson noise can be used in this system.
5.2 Different Noise Types

Figure 5.18: Poisson coding: without pre-processing

<table>
<thead>
<tr>
<th>No.</th>
<th>Text</th>
<th>After Subtract Cover</th>
</tr>
</thead>
<tbody>
<tr>
<td>e-version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s-version</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.5: Extracted watermarks after Poisson coding.
5.2 Different Noise Types

5.2.4 Rayleigh Noise

The PDF of Rayleigh noise is given by

\[ p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases} \]

The mean and variance of this density are given by

\[ \mu = a + \sqrt{\pi b/4} \]

and

\[ \sigma^2 = \frac{b(4-\pi)}{4} \]

The basic shape of this density is skewed to the right. This model can be quite useful for approximation skewed histograms. A plot this function and its histogram are shown in Figure 5.19 (a) and (b) respectively. The power spectrum of this noise is flat. The resultant watermarked voucher is shown in Figure 5.20, and the extracted watermarks are shown in Table 5.6. The results show that Rayleigh noise can be used in this system.
5.2 Different Noise Types

Figure 5.20: Rayleigh coding: without pre-processing

<table>
<thead>
<tr>
<th>No.</th>
<th>Text</th>
<th>After Subtract Cover</th>
</tr>
</thead>
<tbody>
<tr>
<td>e-version</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>s-version</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
</tbody>
</table>

Table 5.6: Extracted watermarks after Rayleigh coding
5.2.5 Synthetic Fractal Noise

In [43], a method to create a synthetic fractal is introduced which involves a simple process of filtering white noise of the required size with a low-pass filter determined by the Fractal Dimension $D$. For a one dimension signal, the formula for this filter $Q$ is given by

$$Q(k) = |k|^{-\beta/2} , \text{where } \beta = 5 - 2D$$

and for a two-dimension signal is

$$Q(k_x, k_y) = |k|^{-\beta/2} \text{ where }$$

$$|k| = \sqrt{k_x^2 + k_y^2} \text{ and } \beta = 8 - 2D$$

where $D$ is the fractal dimension. The process of creating a fractal signal or a fractal landscape is described in the following five-stage algorithm

Algorithm 5.3: Generate a fractal noise field

(i) Compute a random Gaussian distributed array $G_i, i = 0 \cdots N - 1$ using a conventional Gaussian random number generator, with zero mean and unit variance.

(ii) Compute a random sequence of uniform distributed numbers $U_i, i = 0 \cdots N - 1$ in the range zero to one.

(iii) Calculate the real and the imaginary parts; $N_i = G_i \cos(2\pi U_i)$ and $M_i = G_i \sin(2\pi U_i), \text{ Thus defines } G_i\text{ as the amplitude and } U_i \text{ as the phase.}$

(iv) Filter $N_i, M_i$ with $W_i = \frac{1}{|k|^{\beta/2}}$ to create $N'$ and $M'$. For surfaces: $|k| = \sqrt{k_x^2 + k_y^2}$ where $\beta = 8 - 2D$ and for signals $\beta = 5 - 2D$

(v) Invert the result using a IDFT to obtain $n_i = \text{Real}(\text{IDFT}(N' + iM'))$. 
A plot of this function and its histogram are shown in Figure 5.21 (a) and (b) respectively. The power spectrum of this noise is not uniformly distributed (see Figure 5.21 (c)).

The resultant watermarked voucher is shown in Figure 5.22, and the extracted watermarks are shown in Table 5.7. The results show that the fractal noise is not suitable in this system. This is due to the non-uniform power spectrum associated with fractal noise. For electronic versions, the post-processing succeeds in extracting the watermark while post-processing can not extract a visible watermark for the scanned version. This is due to the large amount of added noise and errors in the scanned document (especially those related to translation).
5.2 Different Noise Types

Figure 5.22: Fractal coding: without pre-processing.

<table>
<thead>
<tr>
<th></th>
<th>No Text</th>
<th>After Subtract Cover</th>
<th>Post Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>e-version</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
<tr>
<td>s-version</td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Table 5.7: Extracted watermarks after fractal coding.
5.2 Different Noise Types

Figure 5.23: Speckle noise (a) Texture, (b) Histogram, (c) Power spectrum.

5.2.6 Speckle Noise

Generating speckle patterns is a similar process to that of producing fractal noise. Speckle noise is created by low-pass filtering the Fourier transformed image of a Gaussian noise field. A low-pass filter has the property that sets all frequencies greater than $T$ to zero, where $T$ is some bandwidth. The process for creating a speckle noise is described in the following algorithm:
Algorithm 5.4: Generate speckle noise

```
1 I = Speckle(bandwidth) {
2   g = Create a positive Gaussian noise image;
3   G = DFT(g);
4   G' = LowPassFilter (G, bandwidth);
5   g' = IDFT(G');
6   I = Re(g')^2 + Im(g')^2; % intensity image
7 }
```

In general, any filter \( p \) can be applied to the Fourier transform of any source image \( f \). This is the basis of all coherent optical systems which are based on the following convolution formula,

\[
I = |p \otimes f|^2
\]

The image in Figure 5.23 (a) shows a speckle image \((128 \times 128)\) with a frequency cutoff \( = 32 \). The histogram and the power spectrum of the speckle noise are shown in Figure 5.23 (b) and (c) respectively. The resultant watermarked voucher is shown in Figure 5.24. It is clear that the diffusion is not perfect. This is due to the small bandwidth used in the low-pass filter. The extracted watermarks as shown in Table 5.8 which are low-pass filtered as well.

5.2.7 Chaotic Noise

Chaos is often associated with noise in that it is taken to represent a field which is unpredictable [51]. Although this is the case, a signal generated by a chaotic system has more structure if analyzed in an appropriate way. Moreover, this structure often exhibits features that are similar at different scales which leads to a natural connection between the behaviour of chaotic systems - the signals they produce - and fractal or self-affine signals.
5.2 Different Noise Types

Figure 5.24: Speckle coding: without pre-processing

<table>
<thead>
<tr>
<th></th>
<th>No Text</th>
<th>After Subtract Cover</th>
</tr>
</thead>
<tbody>
<tr>
<td>e-version</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>s-version</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
</tbody>
</table>

Table 5.8: Extracted watermarks after speckle coding.
5.2 Different Noise Types

Chaotic signals are typically the product of an iterative process of the form $x_{n+1} = f(x_n)$ where the function $f$ is some nonlinear map which depends on a signal or a set of parameters. For example: $x_{n+1} = rx_n(1 - x_n)$, is a simple quadratic iterator known as the logistic map and has a range of characteristics depending on the value of $r$. One modification of the logistic map is the Matthews cypher:

$$x_{n+1} = (1 + r) \left(1 + \frac{1}{r}\right)^2 x_n(1 - x_n)^r, \ r \in (0, 4]$$

Pseudo-chaotic numbers are, in principle, ideal for coding because they produce number streams that are ultra-sensitive to the initial value (the key). However, instead of using iterative-based maps using modular arithmetic with integer operations (such as those used to generate conventional noise fields), here we require the application of principally non-linear maps using floating point arithmetic. Thus, a drawback concerning the application of deterministic chaos is the processing speed.

The phase distributions for the chaotic field have specific characteristics showing that the phases of the chaotic field are not uniformly distributed unlike those of a Gaussian noise field which are uniformly distributed. The phase distribution
5.2 Different Noise Types

of a phase map is a useful method of quantifying a chaotic signal as it typically provides an unambiguous and unique signature which is specific to a given chaotic system. Figure 5.25 (b) shows the PDF for the Matthew map which reveals a non-uniform distribution. However, the mid range (i.e. for \( x_n \in [0.3, 0.7] \)) is relatively flat, indicating that the probability for the occurrence of different numbers generated by the logistic map in the mid-range is the same. A plot of the of this function and its histogram are shown in Figure 5.26 (a) and (b) respectively.

The approach to using deterministic chaos for coding has to date been based on using conventional and other well known chaotic models such as the Matthews map. However, in watermark diffusion, the physical model from which a chaotic map has been derived is not important; only the fact that the map provides an operator that is 'good' at scrambling the watermark. These points leads to an approach which exploits three basic features of chaotic maps: (i) they increase the complexity of the system; (ii) there are an unlimited number of maps of the form \( x_{n+1} = f(x_n) \) that can be literally 'invented' and then tested for chaoticity to produce a database of algorithms, (iii) uniformly distributed chaotic noise is not generated using the conventional way, instead it is generated using chaotic maps. This provides a new level of security for those systems that are based on uniform
5.2 Different Noise Types

Figure 5.27: Chaos coding, without pre-processing noise for diffusion.

<table>
<thead>
<tr>
<th>No Text</th>
<th>After Subtract Cover</th>
</tr>
</thead>
<tbody>
<tr>
<td>e-version</td>
<td>![Image]</td>
</tr>
<tr>
<td>s-version</td>
<td>![Image]</td>
</tr>
</tbody>
</table>

Table 5.9: Extracted watermarks after chaos coding.

Figure 5.27 shows a watermarked voucher that has been generated using uniform chaotic noise. The extracted watermarks are shown in Table 5.9. The results
show that uniform chaos noise can be used in this system. However, the floating point operations required to implement chaotic fields requires greater computational time.

5.3 Colour Document Coding

In this section, the watermarking methods are extended to colour documents. Figure 5.28 and 5.30 show a coded document. Since colour images consist of three channels (red, green and blue), the diffused watermark can be added to each channel. There are no changes required to the algorithms related to this system, except that each process is repeated for each channel (e.g. the correlation processes is undertaken for each channel).

The diffusion operator used here is the Gaussian operator. The extracted watermarks for electronic and colour-scanned documents are shown in Figure 5.29 (a) and (b) respectively. One interesting point is that the grey-mode scanning is sufficient to extract the watermark, as shown in Figure 5.29 (c).

To generate the coloured background document shown in Figure 5.30, the red channel is only used and the remaining channels are cleared. The extracted watermarks for the electronic version, coloured scanning, and grey scanning are shown in Figure 5.31 (a), (b) and (c).
The Royal Environmental Health Institute

Advanced Food Hygiene Course

Pass

This is to Certify that

Nick James

has attended the course in Advanced Food Hygiene organized by the Institute, and, after Examination, has been awarded this Diploma.

this Diploma is granted:

this third day of October 19. 97

President of the Institute

Course Organizer

Certificate No ... 94.094.... Verma. Amman....

Figure 5.28: Coloured text coding.
5.3 Colour Document Coding

Figure 5.29: Extracted watermark from coloured text document: (a) Electronic version, (b) Coloured-scanned version, (c) Grey-scanned version.

Figure 5.30: Coloured background coding.

Figure 5.31: Extracted watermark from coloured background document: (a) Electronic version, (b) Coloured-scanned version, (c) Grey-scanned version.
5.4 Why does the System Work?

There are a number of factors that help to make this system operate successfully in extracting the watermark after scanning with low resolution. These factors are described in the following subsections:

5.4.1 Compatibility with Optical Imaging Systems

The compatibility between the optical image formation and the watermark diffusion is a most important factor. Referring to Section 3.4.1,

For perfectly incoherent illumination, an optical system is linear in intensity and, if isoplanicity holds, the output (image) intensity is equal to the input (object) intensity convolved with the intensity point spread function.

Due to the fact that the scanner is an incoherent optical system, the image produced can be represented by.

\[ S = P \otimes \otimes I \]  

(5.1)

where \( I \) is the input (hard copy) image, \( P \) is Point Spread Function of the scanner and \( S \) is the scanned image (electronic copy). Given that \( I \) is the printed watermarked cover that has been designed using the diffusion model described here, then \( I \) is given by

\[ I = n \otimes \otimes w \]  

(5.2)

where \( n \) is the noise operator used in coding and \( w \) is the watermark (ignoring the added text). By substituting Equation 5.2 in Equation 5.1 we get:

\[ S = P \otimes \otimes n \otimes \otimes w \]  

(5.3)

In the extraction step, the scanned image \( S \) is correlated with the noise field that was used to diffuse the watermark, i.e.
5.4 Why does the System Work?

\[ w' = n \odot \odot S \]

and using Equation 5.3 we have

\[ w' = n \odot \odot P \odot \odot n \odot \odot w \]

By using the linearity property of the convolution operator,

\[ w' = P \odot \odot n \odot \odot n \odot \odot w \]

\[ = P \odot \odot w + d \]

where \( d \) is some added noise. Hence, the extracted watermark for a scanned document has more noise than that extracted from an electronic version. There are two sources of noises: the PSF of the scanner and \( d \). One important source of \( d \) is the noise arising from power spectral regularization or otherwise.

5.4.2 Computational Aspects

Correlation Instead of Registration

Using correlation in the extraction phase, increases the robustness of the system to some important attacks such as translation and cropping (most likely to occur during scanning). This is because correlation is a matching process which does not rely on the exact location of each pixel. The correlation of \( p \) and \( c \) will be maximum (i.e. produces a local maximum or spike) at the location where \( p \) finds a correspondence in \( c \). On the other hand, algorithms that depend on the spatial location must be registered to each other before the algorithm can be executed. For example, the subtraction (or division) between two images, where misalignment between them needs to be corrected first. Registration adds more complexity to the system. It involves determining the transformation which will map pixels in one image to their corresponding or matching pixels in another image.
5.4 Why does the System Work?

Watermark Visibility and Robustness

Most of the literature has focused on invisible digital watermarks which have wider applications. However, in general, any increase in robustness comes at the expense of increased watermark visibility. In this kind of application (document watermarking), the diffused watermark can be visible (background) without affecting the fidelity of the document. This visibility increases the robustness of the system, even after a print/scan attack.

Noise as a Diffusion Operator

The presence of any pattern in the diffusion operator is likely to make the watermark clear after diffusion. For example, even though there is no obvious pattern in speckle noise, the watermark can be recognized after diffusion (see Figure 5.24). On the other hand, the watermark cannot be recognized after uniform coding (see Figure 5.14). Moreover, by using noise, the diffused watermarks is statistically invisible with good auto-correlation properties. Not only is the unpredictability of the field required to make the watermark robust, it is the first line of defense against an attack [53]. Hence, noise field diffusion provides an optimum approach for watermarking in terms of the fidelity of the extraction process, robustness and resistance to attack.

Key Dependent System

Noise generation depends on the seed which can play the role of the key. Only an exact match recovers the watermark; all other combinations generate a noise-like signal. The attacker must know a significant amount of information before they can break the system, such as:

(i) The correct seed.
5.4 Why does the System Work?

(i) The diffusion operator type (i.e. Gaussian, Uniform, etc).

(ii) The original image size (The scanned image must be scaled to its original size before extraction).

(iv) The function used to generate a chaotic noise field (if appropriate)

Moreover, using uniform chaos can mislead the attacker, guiding him/her to a conventional uniform noise attack.
Chapter 6

Attack and Robustness Analysis

In practice, a watermarked document may be altered either intentionally or unintentionally but a watermarking system should still be able to detect and extract the watermark. The distortions are limited to those that do not produce excessive degradations otherwise the transformed object is unusable. Examples of processes that a watermark might need to 'survive' are lossy compression, filtering, and geometric distortion.

In this chapter, in order to evaluate the proposed watermarking technique, the system is tested against some important attacks. All experiments are applied on the watermarked test document that is shown in Figure 6.1(a) \((277 \times 388)\). The original and extracted watermarks \((60 \times 60)\) are shown in Figure 6.1 (b) and (c) respectively. The diffusion is undertaken using a Gaussian noise field with a seed of 10.

To measure the effect of the attacks on the extracted watermark, pixel-based visual distortion metrics are used. The quantitative distortion metrics allow for fair comparison between different images. One way to test the similarity between
5.4 Why does the System Work?

Figure 6.1: Attack experiment parameters: (a) Watermarked document used to test attacks, (b) Original watermark, (c) Extracted watermark.
two images is the Normalized Mean Square Error:

$$\text{NMSE} = \frac{\sum_{ij} (P_{ij} - Q_{ij})^2}{\sum_{ij} (P_{ij} + Q_{ij})^2}$$

where $P_{ij}$ represents a pixel, whose coordinates are $(i, j)$, in the original undistorted image, $Q_{ij}$ represents a pixel, whose coordinates are $(i, j)$ in the distorted image. If the two images are identical, NMSE returns zero. If the difference between the two images are significant, then NMSE returns a number close to one.

### 6.1 Lossy Compression

Lossy compression techniques try to reduce the amount of information by removing imperceptible signal components. The loss of information can be acceptable because the computer representation of the signal contains redundancy with respect to what is needed for human perception and interpretation.

One of the most popular forms of lossy compression is the JPEG method. The compression itself is performed in three sequential steps: block-DCT computation, quantization, and variable-length code assignment [24]. The ‘Quality’, a number between 0 and 100, is usually bounded with this kind of compression. Higher numbers imply better quality (i.e., less image degradation due to compression), but the resulting file size is larger.

Due to the fact that JPEG is designed to disregard redundant perceptually insignificant information, and that our watermark is a visibly diffused watermark inserted into the background, it is expected that this coding system is robust to JPEG. In this experiment, the watermarked document is compressed using different quality values. The extracted watermarks are compared to the original watermark, and the results are plotted in Figure 6.2 (a). Extracted watermarks for quality = 80, 50 and 20 are shown in Figure 6.2 (b). The results show that the
6.1 Lossy Compression

![Graph](image)

**Figure 6.2**: JPEG test results: (a) Results of JPEG test, (b) Extracted watermark for JPEG quality = 20, 50 and 80 (from left to right).

The method is robust for JPEG even with a very low quality.

It is very important that the watermarking technique is robust to JPEG, since most images are vulnerable to compression, especially if they are to be uploaded to the Internet, when the scanner (or any image capture device) and the extractor program are in two different locations and the Internet is used to transfer the images.
6.2 Linear Filtering

Another common type of process that changes images in a deterministic fashion is linear filtering, i.e.

$$I' = I \otimes f$$

where $I$ is an image, $f$ is a filter, and $\otimes$ denotes 2D convolution. Many normal operations on images are explicitly implemented with linear filters. The blurring and sharpening effects in image editing programs apply simple filtering operations [1]. In addition, a scanning process can be modelled by convolution.

To understand the effect of linear filtering on the system, the following analysis is provided: Convolution in the spatial domain corresponds to a multiplication in the Fourier domain. Thus, we can think of each Fourier component as being either attenuated or amplified by a real number. A diffused watermark can be represented in the frequency domain using polar coordinates by

$$\text{PW} = A_p e^{i\theta_p} A_w e^{i\theta_w}$$

where $A_p$ is the amplitude spectrum of the diffused operator, $A_w$, is the amplitude spectrum of the watermark, $\theta_p$ and $\theta_w$ are the phase spectra of the diffused operator and the watermark respectively. The filtered diffused watermark can be represented by

$$\text{FPW} = A_f e^{i\theta_f} A_p e^{i\theta_p} A_w e^{i\theta_w}$$

where $A_f$ and $\theta_f$ are the amplitude spectrum and the phase spectrum of the filter respectively. During watermark extraction (i.e. correlation), we get

$$W' = A_f e^{i\theta_f} A_p e^{i\theta_p} A_w e^{i\theta_w} A_p e^{-i\theta_p}$$

$$= |A_p|^2 A_f e^{i\theta_f} A_w e^{i\theta_w}$$

Hence, the extracted watermark is a filtered version of the original watermark. In other words, in the frequency domain, the watermark is spread across the whole
range of frequencies (*Spread Spectrum Coding*). Thus, if the watermarked document is distorted by some process that affects only a fraction of the frequency spectrum, as with linear filtering, then the watermark is still identifiable. This result is supported by the following experiments.

![Graph](image)

**Figure 6.3**: GLPF test results: (a) Results of GLPF test, (b) Extracted watermark for GLPF with sigma = 20, 50 and 80 (from left to right)

### 6.2.1 Low-Pass Filtering

To test the effect of low-pass filtering, the watermarked document is distorted with Gaussian Low-Pass Filter (GLPF) of varying degrees of the standard deviation sigma (set to radu values). The extracted watermarks are compared to the original and the results are plotted in Figure 6.3 (a). Extracted watermarks for sigma = 80, 50 and 20 are shown in Figure 6.3 (b). The results show that the method is
6.2 Linear Filtering

robust for GLPF, even for very small values of sigma.

![Graph](image)

Figure 6.4: IHPF test results: (a) Results of IHPF test, (b) Extracted watermark for IHPF HBW = 20, 50 and 80 (from left to right).

6.2.2 High-Pass Filtering

In this test, the watermarked document is distorted with an Ideal High-Pass Filter (IHPF) of varying Half-Bandwidth (HBW). The difference between the extracted watermark and the original are plotted in Figure 6.4 (a) Extracted watermarks for HBW = 20, 50 and 80 are shown in Figure 6.4 (b). The results show that the method is robust for IHPF even with large HBW (the watermark can be recognized from its edges).
Many processes that may be applied to an image have the effect of additive noise [1]. That is,

\[ c_n = c + \alpha n \]

where \( c \) is the original image, \( n \) is the noise, and \( \alpha \) is the embedding strength. For example, Gaussian noise arises in an image due to factors such as electronic circuit noise and sensor noise due to poor illumination and/or high temperature [24]. Such noise processes are cases of additive noise. Most watermarking techniques are specifically designed to survive this type of distortion. In this test, the effect of additive noise on the system is investigated. The watermarked document is

![Figure 6.5: Additive noise test results. (a) Results of additive noise test, (b) Extracted watermark for embedding strength (%) = 20, 50 and 80 (from left to right).](image-url)
6.4 Amplitude Effects

distorted with additive Gaussian noise \((seed = 100)\) with different embedding strengths \(\alpha\). The extracted watermarks are compared to the original watermark and the results are plotted in Figure 6.5 (a). Extracted watermarks for \(\alpha = 0.20\), 0.50 and 0.80 are shown in Figure 6.5 (b). The results show that the method is robust for additive noise. Note that if the same noise field as that used for diffusion with \(\alpha = 1\) is applied, then the extracted watermark will be a spike.

6.4 Amplitude Effects

Many processes applied to a watermarked document are deterministic functions. A simple, but important example is the change in amplitude, i.e.

\[ c_n = uc \]

where \(c\) is the original image and \(u\) is a scaling factor. For music, this simply represents a change of volume. In an image, it represents a change in brightness and contrast. In this experiment, the contrast of the watermarked document was modified by scaling the amplitude of the image by several scaling factors between \(u = 1\) and \(u = 0\). Extracted watermarks for \(u = 20, 50\) and \(80\) are shown in Figure 6.6 (a), (b) and (c) respectively. This experiment shows that 'amplitude effects' have no influence on the model, because any change in the contrast can be reversed by removing the foreground with a suitable threshold.

Figure 6.6: Amplitude change test results (a) Extracted watermark for scale = 0.2, (b) Extracted watermark for scale = 0.5, (c) Extracted watermark for scale = 0.8.
6.5 Thresholding

A thresholded image \( g(x, y) \) can be defined as

\[
g(x, y) = \begin{cases} 
1 & \text{if } f(x, y) > T \\
0 & \text{if } f(x, y) \leq T
\end{cases}
\]

where \( f(x, y) \) is the original image and \( T \) is the threshold. This is the simplest of all thresholding techniques where we partition the image histogram by using a single global threshold \( T \). Image thresholding plays a central role in the applications of image segmentation. Segmentation subdivides an image into its constituent regions or objects. In our system, thresholding can be applied on the scanned watermarked document (i.e. instead of scanning the document using grey levels). Binary scanning is faster than grey-level scanning. Note, that the 'Foreground Removing' option does not have to be executed after thresholding. The experiments show that the watermark can be extracted after thresholding in a specific range (\( 65 \leq T < 95 \)). Most of the data outside this range are 'cleared' (i.e. set to one or zero). The experimental results are shown in Figure 6.7. Figure 6.8 shows the watermarked document after being thresholded with \( T = 0.75 \). The Figure shows the thresholding interface used in this system.

6.6 Geometric Distortion

Geometric distortion affecting image data includes rotation, spatial scaling, translation, and changes to the aspect ratio. Although many different approaches have been investigated, robustness to geometric distortion remains one of the most difficult outstanding areas of watermarking research [54]. The geometric distortions that are applied to the watermarked document must be identified first. Then the distortion is inverted before the extraction proceeds. Most suggested approaches fall into one of the following categories:
Figure 6.7: Thresholding test results: (a) Results of thresholding test, (b) Extracted watermark for threshold = 0.60, 0.65, 0.75, 0.85 and 0.95 (from left to right)
(i) Exhaustive search: After defining a range of likely values for each distortion parameter, every combination of parameters is examined.

(ii) Registration: when the original non-watermarked document is available at the extractor side, regions suspected of containing a watermark are aligned (using techniques from the pattern recognition literature) prior to a single application of the extractor. A common approach for public extraction is the embedding of a dedicated synchronization pattern in addition to the embedded watermark [55].

(iii) Autocorrelation. The autocorrelation result (i.e., peaks in the autocorrelation surface) can be used to identify and invert geometric distortion.

(iv) Invariant watermarks: Rather than detect and invert the geometric distortions, an alternative approach is to design watermarks that are invariant to such distortions [35].

In the following subsections, our system is tested against these kind of distortions.
6.6 Geometric Distortion

6.6.1 Scaling

The size of any image captured by a scanner depends on the resolution used. The size of the image is usually different from the original because the resolution in the scanner and the printer may be different. Since scaling in the spatial domain causes inverse scaling in the Fourier domain (i.e., as the spatial scale expands, the frequency scale contracts and the amplitude increases linearly), the scaling must be reversed before extracting the watermark. The original document size is part of the key (see Section 4.6.2), hence it can be used to rescale the document to its original size before applying the correlation since correlation is not a scale invariant process. Figure 6.9, shows the extracted watermarks after: (a) scaling up to 348 x 488 while preserving the aspect ratio, (b) scaling down to 206 x 288 while preserving the aspect ratio. (c) and (d) scale to 200 x 400 and 200 x 300 respectively without retaining the aspect ratio. In all cases, the watermark can be extracted.

6.6.2 Cropping

Cropping is a typical effect of scanning. The scanned image may be cropped (i.e., to include only a part of the original image). Figure 6.10 and Figure 6.11 show two examples of cropped documents and the corresponding extracted watermarks. The test shows how robust the system is to cropping. Furthermore, Section 5.1.5
6.6 Geometric Distortion

Figure 6.10: Cropping test results: (a) Cropped document (top=70, left=50, height=199, width=299), (b) Extracted watermark after cropping

Figure 6.11: Cropping test results (a) Cropped document (top=0, left=100, height=288, width=277), (b) Extracted watermark after cropping.

shows that the system is also robust to irregular cropping. This robustness is justified by the following analysis

Redundant Effects of Convolution

Physically, convolution can be thought of as a ‘blurring’ or ‘smearing’ of one function (image) by another. This blurring effect is clear from the definition of convolution. Consider the following definitions of convolution.
6.6 Geometric Distortion

(i) Using Fourier transform, the convolution between two functions is the same as the product of their Fourier transforms in Fourier space (*convolution theorem*).

(ii) Mathematically, the convolution between two images of size $M \times N$ is defined by the following expression [24]:

$$f(x,y) \otimes h(x,y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)h(x-m, y-n) \quad (6.1)$$

Equation 6.1 is just an implementation of (i) flipping one image (mask) about the origin, (ii) sliding that function (image) past the other by changing the values of $(x, y)$; (iii) at each displacement $(x, y)$, the entire summation in Equation 6.1 is carried out.

Convolution can be thought of in terms of a *Redundant Embedding* of one image with another. To understand this, it is helpful to consider the definition of the impulse function. An impulse function of strength $A$, located at coordinates $(x_0, y_0)$, is denoted by $A\delta(x-x_0, y-y_0)$ and is defined by the expression

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} s(x,y) A\delta(x-x_0, y-y_0) = As(x_0, y_0) \quad (6.2)$$

This equation represents one of the most important properties of the impulse function, namely, the shifting property. Equation 6.2, and using the definition of convolution (Eq 6.1), states that the convolution of a function with an impulse *copies* the value of the function (multiplied by the strength of the impulse) at the location of the impulse [24]. Figure 6.12 shows two impulse images. The top left image contains an impulse located at $(1,1)$ with strength $A = 1$ (Part of the impulse image is magnified beside it). The bottom left image contains an impulse located at $(20,20)$ with strength $A = 0.5$. The convolution of these two impulses with the 'Loughborough logo' illustrates the effect of the shifting property of the impulse function. The right image in Figure 6.12 (R1) shows the addition of these two images. In Figure 6.13, the left image has the two spikes used in in Figure
6.6 Geometric Distortion

Figure 6.12: Redundant effect of convolution: The convolution is simulated using addition in the spatial domain.

6.12. The right image (R2) in Figure 6.13 shows the convolution of this image with the Loughborough logo. Note, that the result of the two experiment R1 and R2 are identical.

Since any image \((M \times N)\) used for diffusion is a set of \((M \times N)\) impulses with different strength, then the convolution between this image and the watermark can be thought of as an addition of \((M \times N)\) images. Each image is a shifted version of the watermark with a strength equal to the corresponding impulse. The addition of images in the spatial domain is equivalent to addition in Fourier space (i.e. The DFT is a linear operator).

From the previous analysis, the texture produced from diffusion is nothing more than a redundant embedding of the watermark. This redundancy helps this system to be robust against cropping attack.
6.6 Geometric Distortion

Figure 6.13: Redundant effect of convolution: The convolution is done in the frequency domain.

6.6.3 Spatial Translation

A translation or shift implies zero padding of the image, such as would occur if an image were placed on a scanner and scanned (i.e., scanning the whole image with an additional background). Zero-padding in the spatial domain is equivalent to increasing the sampling resolution in the frequency domain. Figure 6.14 shows the extracted watermark after applying different translation shifts on the watermarked document. Watermarks (a), (b), (c) and (d) are extracted after zero-padding the watermarked document. The remaining images (i.e., e, f, g, and h) are extracted from a watermarked document that was padded by 1s. The shifting is undertaken as follows:

- (a) and (e): top = 10, left = 10, bottom = 10 and right = 10 pixels.
- (b) and (f): top = 1, left = 1, bottom = 1 and right = 1 pixels.
- (c) and (g): top = 10, left = 10, bottom = 0 and right = 0 pixels.
- (d) and (h): top = 0, left = 0, bottom = 10 and right = 10 pixels.

Cropping the translated document to its original size is important before applying the correlation. Figure 6.15 shows the extracted watermark after the tested document has been shifted by 10 pixels from top and left margin.
6.6 Geometric Distortion

Figure 6.14: Extracted watermark after translation: (a, b, c and d) from the zero-padded watermarked document, (e, f, g and h) from one-padded watermarked document.

Figure 6.15: Translation test results without cropping.

6.6.4 Rotation

Almost all applications involving printing and scanning will result in some degree of rotation. On the other hand, robustness to rotation is still one of the most difficult forms of attack. Most watermarking algorithms are robust to a small degree of rotation while they fail to extract the watermark for large degrees of rotation. If document verification is based on a flat-bed scanner, it is possible to place the image in the corner of the scanner in order to minimize the effect of rotation.

Rotating an image by an angle \( \theta \) in spatial domain rotates the frequency space data by the same angle. Hence, in our system, the rotation must be reversed before applying the correlation. Figure 6.16 shows the extracted watermarks for documents that have been.
6.6 Geometric Distortion

- a) Rotated by one degree.
- b) Rotated by one degree and then cropped to its original size.
- c) Rotated by one degree, rotation inverted and then cropped to its original size.
- d) Rotated by two degrees.
- e) Rotated by two degrees and then cropped to its original size.
- f) Rotated by two degrees, rotation inverted and then cropped to its original size.

This experiment shows that the system can not extract the watermark after rotation, unless, the tested document was rotated by one degree or less ($\theta \leq 1$) and then cropped to its original size. For larger angles, the rotation must be reversed first and then cropped to its original size. This 'Reverse Process' can be applied using any of the techniques mentioned previously (autocorrelation, etc). In this system, an interactive interface (see Figure 6.17) has been designed to help the user.
test several angles and see the results directly. This interface can be used to test the remaining geometric distortion.

In the following chapter, some applications that can benefit from this system are introduced.
Figure 6.17: Geometric distortion interface.
Chapter 7

Applications, Future Work and Conclusions

7.1 Applications

The watermarking method discussed in this thesis can be used in wide variety of applications. In this chapter, some of these applications are demonstrated.

7.1.1 Authentication

Authentication of a document image should ensure that the document has not been altered from the time it was created and signed by the author to the time it was received at the destination (i.e. exact authentication). Authentication of paper documents is an important concern as the ability of counterfeiters has been increased substantially in recent years. This is contributed to by the dramatic improvement in the capability of high resolution scanners and printers. Moreover, digital documents can be accessed and modified by intruders relatively easily. This is especially true in the case of documents that are exchanged over the Internet.
Figure 7.1 shows a sample of an ID card, where a grey, scaled down version of the photo has been diffused (using option 2 from the adaptive filtering) and printed beside the original image. Substantial editing, such as changing the original photo, will be illegitimate because it will completely change the interpretation of the card. Thus, a photo verification system can be designed to do the following:

(i) Capture the diffused watermark using any tool (scanner, camera, etc)

(ii) Read the key. The key might be:

   (a) Encoded using bar code, or

   (b) Stored in a local Database, or
7.1 Applications

Figure 7.2: Watermark: combination of photo and text.

(c) Stored in a distance Database that can be accessed via the Internet.

(iii) Extract the watermark.

(iv) Verify the authenticity by comparing the original photo with the extracted one. This can be done either by:

(a) A subjective test, using the judgment of human beings. For more details on the scales that have been suggested for use in evaluation of watermarking quality refer to [56].

(b) Quality metrics, such as the Mean Square Error (MSE).

(c) Any other matching algorithm including the application of an Artificial Neural Network.

Such a system can be modified to include more information in the diffused watermark, such as the name of the ID card holder (see Figure 7.2). Moreover, the diffused watermark can be the background texture for the ID card.

Statistical Verification System

When a document is prepared using MS Word, for example, or any other major word processing package, statistical information from the document can be
7.1 Applications

Figure 7.3: Coded statistical information

gathered, information about the author, date and time, number of characters and spaces and so on. A verification system can use this information to check the authenticity of the document. Any attempt at modification of the file will be reflected in its statistics. The system can either incorporate these data in plain text or as a diffused code into a patch on the document which is encoded into an indecipherable image as shown in Figure 7.3. The image needs to be attractively packaged in an appropriate place on the document - the bottom right hand corner or example.

It is assumed that the recipient of the document (scanned or electronic) will have the appropriate software available. The encoded image is read into the decoding software and text-recognition used to reveal the text which is then compared with the plain-text statistics of the document. The data in the image can alternatively be checked manually against the statistics of the file instead of using text recognition. Each author can have a particular key for encoding of image. Upon receipt, the recipient applies that particular key to decode the image. Alternatively, a separate one-time PIN can be transmitted to the recipient in order to decode the image.

Specific Parts Verification System

The system described in the previous section can be extended to include a ‘specific parts’ from of the text that must be correct, e.g. a sum of money, name of
Bank X with sort code 12-23-34 will transfer the sum of $2,500 from account number 9876543 to account number 12345678 at Bank Y with sort code 98-76-54

| Bank X with sort code 12-23-34 will transfer the sum of $2,500 from account number 9876543 to account number 12345678 at Bank Y with sort code 98-76-54 |
|---|---|---|
| 12-23-34 | $2,500 |
| 9876543 | 12345678 | 98-76-54 |

Figure 7.4: Coded specific part from a document.

Figure 7.5: Revealed document after coding specific parts.

beneficiary etc. An example is shown in Figure 7.4. After decoding, the message would be as given in the Figure shown in 7.5.

Clearly, the diffused code could be placed into the background of each data field (i.e. instead of placing it in the next empty line).

Another example, Figure 7.6(a) shows a gift voucher that has been used in the previous chapter (see Figure 6.1). In this voucher, the amount has been edited and changed from 500 to 5000. The counterfeiting is easily discovered when the watermark is extracted, as shown in Figure 7.6(b).

7.1.2 Transaction Tracking

Also called fingerprinting, transaction tracking involves the embedding of a different watermark into each distributed copy. This is especially useful for identifying people who obtain a document legally but illegally redistribute it.
Leaked Document Monitor

One common method to monitor and discover any ‘leak’ associated with a very important document is to use visible marks. For example, highly sensitive documents are sometimes printed on backgrounds containing large grey digits using a different number for each copy. Records are then kept about who has which copy. Of course, imperceptible watermarks (or at least diffused watermark) are preferable to visible marks (Visible marks are easy to remove/replace from a document when it is copied). Using this model for document watermarking, the tracking number is diffused and inserted into the background (see Figure 7.7). The diffused watermark is inseparable from the document. The adversary (a person who attempts to remove, disable, or forge a watermark for the purpose of circumventing its original purpose) does not know the embedded number and can not recognize the difference between copies (it is difficult for human eyes to find a difference between two copies with different watermarks) Note, that a colour background can be generated by adjusting with the colour balance in each channel (i.e. RGB chan-
Further, each RGB channel can be based on a different key. The extracted watermark is shown in Figure 7.8

7.1.3 Owner Identification (Copyright)

Copyright can be undertaken by embedding the identity of a document's copyright holder as a watermark in order to prevent other parties from claiming the copyright of the document. The embedded data can be a biometric characteristic (such as signature). The receiver of the document reconstructs the signature used to watermark the document, which is then used to verify the authors claimed identity.

Signature Verification System

Handwritten signatures are commonly used to certify the contents of a document or to authenticate legal transactions. A handwritten signature is a well-known
biometric attribute. Other biometric attributes, which are commonly used for authentication include iris, hand geometry, face, and fingerprints (See [57, 58]). While attributes like the iris and fingerprints do not change over time, they require special and relatively expensive hardware to capture the biometric data. An important advantage of the signature over other biometric attributes is that it has been traditionally used in authenticating documents and hence is socially accepted. Signature verification is usually done by visual inspection. In automatic signature verification, a computer takes over the task of comparing two signatures to determine if the similarity between the signatures exceeds some pre-specified threshold. There are many similarity measures that can be used for this purpose. Figure 7.9(left) shows an example for this approach. The signature of the customer is diffused and inserted into the background of the cheque. Each customer has their own key that is known only to them and their bank. They use the key to generate the background and then print the cheque. The bank then uses the key to extract the customer signature from the cheque. If the extracted and the existing signatures on the cheque are matched to each other, then the cheque is accepted.
Figure 7.9: (Left) watermarked cheque, (Right) extracted watermark.
7.1.4 Steganography and Cryptography.

In steganography, one message is hidden inside another, without disclosing the existence of the hidden message or making it apparent to an observer that this message contains a hidden message [2]. Moreover, the information hidden by a watermarking system is always associated with the object to be protected or its owner while steganographic systems just hide information. On the other hand, cryptography can be defined as the study of secret writing (i.e. concealing the contents of a secret message by transforming the original message into a form that cannot be easily interpreted by an observer).

Clearly, our model (diffusion and confusion) can be easily used in both applications. The hidden message can be transformed into a diffused form (i.e. encrypted) and inserted into the background. The hidden information might have no relation with the text (foreground). At the same time, backgrounds are usually used with documents and so diffused data will not necessarily trigger the attention of an observer. Moreover, the hidden message is also encrypted which increases the security level of such documents.

7.2 Future Work

In this section some ideas that require further investigation based on the model and methods developed for this thesis are discussed.

7.2.1 Additive Security Barrier

An additive security barrier is to transfer the diffused watermark to a texture that looks like a 2D barcode (see Figure 7.10). The reason for using this transfer is to increase the security of the system and to force potential attackers to proceed incorrectly in their attempts to break the system.
2D barcodes are two-dimensional graphical pattern that encode information. They allow for higher information density than a standard one-dimensional barcode. A one dimension barcode is a machine-readable representation of information in terms of the widths and spacings of printed parallel lines. They can be read by optical scanners called barcode readers or scanned from an image using special software.

### Generating 2D Barcode Texture

For application to 2D barcodes we consider the following.

(i) The diffused image is a grey-scale image, while the target barcode is a black/white image.

(ii) This watermarking model is robust against thresholding attack (see Section 6.5). Clearly, the robustness can be increased if the watermark is binary and the texture field is used (i.e. without adding a foreground). In this case, the watermark can be extracted even if we set the threshold $T$ to 50% or less.

(iii) XOR is exclusive-or operation ($\oplus$). It’s standard operations on bits are:

\[
\begin{align*}
0 \oplus 0 &= 0 \\
0 \oplus 1 &= 1 \\
1 \oplus 0 &= 1 \\
1 \oplus 1 &= 0
\end{align*}
\]
If one binary image is XORed with a binary uniform noise (where the black pixels are roughly equal to the number of white pixels), then the resultant image has a uniform distribution of white and black points and resembles a 2D barcode.

(iv) XOR is a symmetric algorithm. XORing the same value twice restores the original value \((a \oplus b \oplus b = a)\). In other words, the coding and decoding uses exactly the same program.

Based on the points above, the following algorithm that transfers a grey-scale image to a texture field that resemble 2D barcode (or to transfer a texture field that resemble 2D barcode to a grey-scale image) is suggested.

**Algorithm 7.1: Uniform modulation.**

```
O = UniformMod(I, seed){
    [cols, rows] = size(I);
    U = GenerateUniformNoise(cols, rows, seed);
    T = 50;
    I = Threshold(I, T);
    U = Threshold(U, T);
    O = I \oplus U;
}
```

In terms of a systems protocol, this algorithm can be the last step in the 'coding phase' and the first in the 'extraction phase'. The diffused watermark (Figure 7.11 (a)) is converted into a binary image (Figure 7.11 (b)). Next, a uniform noise is generated (Figure 7.11 (c)) and converted to a binary image (Figure 7.11 (d)) using the same threshold. Finally, the two images (Figure 7.11 (b) and (d)) are XORed with each other (Figure 7.11 (e)). The same algorithm is then used for decoding, except that the input is the 2D barcode texture field. The extracted watermark is
7.2 Future Work

Figure 7.11: Generating a 2D barcode texture field: (a) Grey-scale image; (b) Binary image after thresholding image in (a); (c) Uniform noise; (d) Binary image after thresholding image in (c); (e) 2D barcode; (f) Extracted watermark from 2D barcode.

shown in Figure 7.11(f) It would appear, that the extracted watermark is clear enough, making this approach practically realizable.

A suggested application for this system is shown in Figure 7.12. It shows a part from an official document of Loughborough University where the university logo appears on the top of the document. The Loughborough logo is diffused and placed beside the logo.

7.2.2 A Further Security Barrier

In this application, the diffused watermark is replaced (instead of transferred) with a real 2D barcode. An example is shown in Figure 7.13.

In order to use a published 2D barcode to generate the diffused watermark, a third image must be created and stored somewhere else. The third image is generated by simply XORing the 2D barcode and the binary version of the diffused
7.2 Future Work

Figure 7.12: Document header consisting of a logo and a semi-2D barcode.

Figure 7.13: Document header consisting of a logo and a real 2D barcode.
watermark. The creation of this image is compounded in the following algorithm:

**Algorithm 7.2: Create a third image from the input images.**

```plaintext
I3 = CreateThird(I1, I2)
T = 50;
I1 = Threshold(I1, T);
I2 = Threshold(I2, T);
I3 = I1 \& I2;
```

Here, the 2D barcode plays the role of a public key while the third image is plays the role of the private key. The suggested scenario for the coding step is as follows:

(i) Generate the diffused watermark.

(ii) Determine the public key.

(iii) Generate a 2D barcode that represent the public key.

(iv) Generate the third image from the 2D barcode and the diffused watermark.

(v) Store the third image as a private key (compressed version).

(vi) Publish the 2D barcode as a public key.

The suggested scenario for the decoding step is as follows:

(i) Read the public key from the 2D barcode using any barcode reader

(ii) Use the public key to retrieve the private key.

(iii) Generate the third image (binary diffused watermark) from the 2D barcode and the private key
(iv) Extract the watermark.

In this way, we increase the confusion of the attacker by publishing a real 2D barcode. Moreover, the extracted data from the 2D barcode is nothing but a public key that will be used to get the private key.

**XORing after Scanning**

One drawback of using the XOR processes in the decoding step is: the two images (the 2D barcode and the third image) must be registered with each other pixel by pixel. Normally, a scanned image may have some translation. Consequently, the XOR operation is not robust to scanning unlike the convolution operation. Figure 7.14 shows the extracted watermark after using XOR in coding and decoding. One solution to this problem, is to replace the XOR process by convolution in coding step and correlation (Wiener filter) in decoding step. Figure 7.15 shows the extracted watermark after using convolution/correlation instead of XOR. The comparison between the two results shows that XOR is not suitable for paper watermarking. However, it is simpler to implement than the convolution/correlation process.
7.3 Conclusions

Valuable paper documents are subject to misuse by criminals. This is largely due to the dramatic improvement in personal computer hardware and peripheral equipment. Embedding watermarks into a printed document is one way to secure them. The ability to extract the watermark from a printed copy is generally useful to help establish ownership, authenticity, and to establish the origin of an unauthorized disclosure. However, finding a robust watermarking technique is a continuing challenge. This is due to extensive amount of noise that is added when a document is printed and scanned. Moreover, printed documents do not maintain their quality over time.

In this work, a new watermarking method for paper security is presented. Unlike traditional watermarking techniques, the new approach can extract the hidden watermark after a print/scan attack. This is achieved by using the convolution and correlation processes for coding and decoding respectively. This approach is chosen because of its compatibility with the principles of the physical optics involved in scanning a document. The watermark \( w \) is convoluted (diffused) with a source of noise \( p \) and placed into the background of a text document. The watermark is extracted by removing the text \( n \) using a modified median filter. Then the diffused watermark is correlated with the original noise source. The whole process...
(i.e. coding and decoding) is described in the following formula:

\[
\begin{align*}
    w' &= \text{IDFT} \left\{ \frac{w e^{i\theta_w} A_p e^{i\theta_p} + n}{A_p e^{i\theta_p}} \right\} \\
    &= \text{IDFT} \left\{ A_p w e^{i\theta_w} \right\} A_p^2
\end{align*}
\]

where \( w' \) is the extracted watermark. The extracted watermark is a noisy version of the embedded watermark. This noise is due to the power spectrum \( |A_p|^2 \). In order to enhance the extracted watermark, the power spectrum term must be eliminated or at least minimized its effect. One way to do this is to divide the output over the power spectrum during the diffusion step or correlation step. To avoid singularities, each zero in \( |A_p|^2 \) is replaced by 1. Alternatively, choose \( p \) such that it has a homogeneously distributed power spectrum across all frequencies, (such as white noise) or pre-process the diffusion operator \( p \) by replacing its amplitude spectrum with a constant value. However, these conditions are restrictive.

It has been demonstrated that this method is robust to a wide variety of attacks including geometric attacks, drawing, crumpling and print/scan attack. Experiments also show that the method is relatively insensitive to lossy compression, filtering, amplitude adjustments, additive noise and thresholding. The experiments undertaken also show that the system is 'weak' in terms of a rotational attack. This can be minimized by placing the document in the corner of the flatbed.

The visibility of the diffused watermark and the compatibility of this system with the physical principles of an imaging system, increase the robustness of the system and provide a successful approach to the extraction of the watermark after scanning at low resolution which to date is not possible using any existing commercial system. Furthermore, using correlation in the extraction phase increases
the robustness of the system to some important attacks such as translation and cropping (most likely to occur during a scan).

A variety of applications can benefit from this system, such as photo verification, verification of document statistics, leaked document monitoring and signature verification. The system is secure and can not be attacked easily. First, the feature is 'unsuspicious' as many documents have a background texture. Second, the attacker does not know the algorithm used to generate the diffused watermark. Even if the attacker does know the algorithm, he/she must still know a significant amount of information before the system can be broken, such as: the correct seed, the diffusion operator type, the original image size and so on. Finally, chaos can be used to generate noise fields based on an unlimited number of functions to provide multi-algorithmicity.

Using 2D barcode as a replacement for the diffused watermark can add more features to the system regarding its security. Further work should be undertaken in developing this idea and evaluating it in terms of its potential to provide a source of disinformation.
Appendix A

Diffraction Theory

The method of watermarking developed for this thesis has been aimed at developing a system that can operate using standard off-the-shelf low resolution image capture devices such as a digital camera (WebCam) and/or a flatbed scanner. These hardware devices use incoherent white light to generate an image. In order to utilize such standard image capture devices, a watermarking scheme must be developed that is compatible with the principles of the optical system used. To do this, we must understand the mathematical models associated with the propagation and scattering of light that is used to form an image and in particular, the linear systems theory approach to image formation that is ultimately compounded in a convolution process (the basis for generated a watermark used for this thesis). This result is a direct consequence of applying a scalar theory for modeling the diffraction of light which, for surface scattering, is based on the application of the Kirchhoff boundary (surface) conditions leading to the Kirchhoff theory of (optical) diffraction. This theory provides us with the basic model for how an optical image is formed from which important properties can be inferred and thus, used to design a watermarking system that is robust to existing optical imaging systems.
A.1 Scalar Diffraction Theory

In order to approximately describe optical diffraction, it is reasonable to first adopt a scalar model for light. This type of model is concerned with a monochromatic scalar 'disturbance'

\[ V(r, t) = U(r) \exp(-i\omega t) \]

where \( U \) is the scalar complex amplitude, \( \omega \) is the angular frequency \((=2\pi \times \text{frequency})\), \( t \) is time and \( r \) is a three dimensional vector which in Cartesian coordinates is given by

\[ r = xx + yy + zz \]

In free space \( V \) satisfies the homogeneous wave equation

\[ \nabla^2 V - \frac{1}{c^2} \frac{\partial^2 V}{\partial t^2} = 0 \]

where \( c \) is the velocity of light and \( U \) satisfies the homogeneous Helmholtz equation

\[ \nabla^2 U + k^2 U = 0 \]

where

\[ k = \frac{\omega}{c} = \frac{2\pi}{\lambda} \]

Here, \( k \) is the wavenumber and \( \lambda \) is the wavelength.

Scalar diffraction theory (which essentially stems from solutions to the above equation - subject to careful interpretation of the physical significance of such solutions) should be regarded as a first approximation to optical diffraction.

The observed intensity \( I \) (the observed quantity at optical frequencies) can be taken to be given by (by definition)

\[ I = |U|^2 \]
A.2 Kirchhoff Diffraction Theory

All calculations in this section and the following sections are based on the context of a simple ‘aperture-based’ optical system. The geometry of such system is shown in Figure A.1. In this figure:

- $C$ is the point where the $z$ axis intercepts the plane of the aperture.
- $Q(x, y)$ is the point on the aperture surface used to evaluate the diffraction integral.
- $P(x_o, y_o)$ is the point on the observation screen giving rise to a portion of the diffraction pattern.
- $z_o$ is the distance from the aperture to the screen.
- $n$ is a vector normal to the aperture surface.
- $|r_o|$ is the distance from $C$ to $P$.
- $|r - r_o|$ is the distance from $Q$ to $P$.
- $\theta$ is the angle between $r_0$ and $z$ axis.

Consider a scalar wave field $U$ described by the homogeneous Helmholtz equation

$$(\nabla^2 + k^2)U = 0$$

Let $U_i$ be the field incident on a surface $S$ and introduce the following (Kirchhoff) boundary conditions

$$U = U_i, \quad \frac{\partial U}{\partial n} = \frac{\partial U_i}{\partial n} \quad \text{on} \quad S$$

In the case of an aperture formed by constructing a hole in a screen, the conditions above apply over the plane of the aperture and $U_i$ may be considered to be the field incident on the plane of the screen in the absence of the screen itself.
The Kirchhoff boundary conditions ignore edge effects at an aperture and thus will only be valid for apertures very much smaller than a wavelength. The diffraction theory which stems from a solution to the Helmholtz equation based on these boundary conditions is called the Kirchhoff diffraction theory.

**Green's Function Solution**

Consider the Green's function \( G \) which is the solution to

\[
(\nabla^2 + k^2)G = -\delta(r - r_0)
\]

and given by

\[
G(r \mid r_0, k) = \frac{1}{4\pi |r - r_0|} \exp(ik |r - r_0|)
\]

We can construct two equations.

\[
 GV^2U + k^2 UG = 0
\]

\[
 U V^2 G + k^2 U G = -U \delta
\]

Subtracting these equations and integrating over a volume \( V \) we obtain

\[
 \iiint_V U \delta dV = \iiint_V (GV^2U - U \nabla^2 G) dV
\]
Using the shifting property of the delta function together with Green's theorem, we obtain a solution for the field $U$ at $r_0$, i.e.

$$U(r_0) = \iint_S \left( G \frac{\partial U}{\partial n} - U \frac{\partial G}{\partial n} \right) dS$$

Introducing the Kirchhoff boundary conditions we arrive at the basic Kirchhoff diffraction formula given by

$$U(r_0) = \iint_S \left( G \frac{\partial U_i}{\partial n} - U_i \frac{\partial G}{\partial n} \right) dS$$

This equation is sometimes called the Kirchhoff integral. To compute the diffracted field using the Kirchhoff integral an expression for $U_i$ must be introduced and the derivatives $\partial / \partial n$ with respect to $U_i$ and $G$ computed. Consider the case where the incident field is a plane wave field of unit amplitude (with wavenumber $k \equiv | \mathbf{k} |$, $\mathbf{k} = k / k$). Then

$$U_i = \exp(\mathbf{i}k \cdot \mathbf{r})$$

and

$$\frac{\partial U_i}{\partial n} = \mathbf{n} \cdot \nabla \exp(\mathbf{i}k \cdot \mathbf{r}) = \mathbf{k} \cdot \mathbf{n} \exp(\mathbf{i}k \cdot \mathbf{r}) = \mathbf{k} \mathbf{n} \cdot \mathbf{k} \exp(\mathbf{i}k \cdot \mathbf{r})$$

The calculation of $\partial G / \partial n$ is more complicated

$$\frac{\partial G}{\partial n} = \mathbf{n} \cdot \nabla G$$

and

$$\nabla G = k \frac{\partial}{\partial x} \frac{\exp(\mathbf{i}k \sqrt{(x-x_0)^2 + \cdots})}{4\pi \sqrt{(x-x_0)^2 + \cdots}} + \cdots$$

$$= \frac{1}{4\pi} \mathbf{k} \times \mathbf{G}$$
where

\[ \hat{m} = \frac{\mathbf{r} - \mathbf{r}_0}{|\mathbf{r} - \mathbf{r}_0|} \]

Therefore,

\[ \frac{\partial G}{\partial \hat{n}} = \hat{n} \cdot \hat{m} \left( \mathbf{k} - \frac{1}{|\mathbf{r} - \mathbf{r}_0|} \right) G \]

In most practical circumstances, the diffracted field is observed at distances

\[ |\mathbf{r} - \mathbf{r}_0|, \text{ where} \]

\[ |\mathbf{r} - \mathbf{r}_0| \gg \lambda \]

This condition allows us to introduce the simplification

\[ \nabla G \approx \imath k \hat{m} G \]

so that

\[ \frac{\partial G}{\partial \hat{n}} \approx \imath k \hat{n} \cdot \hat{m} G \]

The Kirchhoff diffraction formula then reduces to the form

\[ U(\mathbf{r}_0) = \imath k \iint_S \exp(\mathbf{k} \cdot \mathbf{r})(\hat{n} \cdot \hat{k} - \hat{n} \cdot \hat{m})GdS \]

### A.3 Fraunhofer Diffraction

Fraunhofer diffraction assumes that the diffracted wavefield is observed a large distance from the screen. The point of observation is in the far field. For this reason, Fraunhofer diffraction is sometimes called diffraction in the 'far field'. Mathematically, it represents an asymptotic solution to the problem posed by the Kirchhoff diffraction formula given above.

The basic idea is to exploit the simplifications that can be made to the Kirchhoff diffraction integral by considering the case when

\[ |\mathbf{r}| \ll |\mathbf{r}_0| \]
As the point \( r \) moves in the domain of integration, the complex exponent describing the Green's function changes rapidly but if \( r_0 >> r \) where \( r_0 \equiv |r_0| \) and \( r \equiv |r| \) then
\[
\frac{1}{|r - r_0|} \sim \frac{1}{r_0}
\]
and
\[
\hat{n} \cdot \hat{k} - \hat{n} \cdot \hat{m} \simeq \hat{n} \cdot \hat{k} + \hat{n} \cdot \hat{r}_0
\]
where
\[
\hat{r}_0 = \frac{r_0}{r_0}
\]
In this case, the Kirchhoff diffraction integral reduces to
\[
U(r_0) \approx \frac{ik\alpha}{4\pi r_0} \int \exp(ik \cdot r) \exp(ik |r - r_0|)dS
\]
where
\[
\alpha = \hat{n} \cdot \hat{k} + \hat{n} \cdot \hat{r}_0
\]
The next simplification that can be made under the condition \( r_0 >> r \) is to the exponent
\[
\exp(ik |r - r_0|)
\]
Now,
\[
|r - r_0| = [(r - r_0) \cdot (r - r_0)]^{\frac{1}{2}}
\]
\[
= [r^2 - 2r \cdot r_0 + r_0^2]^{\frac{1}{2}}
\]
\[
= r_0 \left(1 - 2r \cdot \frac{r_0}{r_0^2} + \frac{r^2}{r_0^2}\right)^{\frac{1}{2}}
\]
Introducing a binomial expansion of this result,
\[
|r - r_0| = r_0 - r \cdot \hat{r}_0 + \frac{r^2}{2r_0} + \cdots
\]
\[
\simeq r_0 - r \cdot \hat{r}_0; \quad r << r_0
\]
the Kirchhoff diffraction integral reduces to
\[
U(r_0) \approx \frac{ik\alpha}{4\pi r_0} \int \exp(ik \cdot r) \exp(-ik \hat{r}_0 \cdot r)dS
\]
We are now in a position to introduce the geometry of the ‘aperture system’ which can be described using the Cartesian coordinates:

\[ r = \hat{x}x + \hat{y}y + \hat{z}z \]

\[ r_0 = \hat{x}x_0 + \hat{y}y_0 + \hat{z}z_0 \]

Here, \( x_0 \) and \( y_0 \) can be taken to describe the position on a flat screen at a distance \( z_0 \) from the diffracting aperture. Consider the following physical conditions.

- \( \hat{k} = \hat{z} \) \( \) The aperture is illuminated by a plane wave at normal incidence
- \( r_0 \approx \hat{z} \) \( \) The diffraction pattern is observed only at small angles
- \( kz \to 0 \) \( \) The aperture is ‘infinitely thin’

The first two conditions give

\[ \alpha \approx \hat{n} \cdot \hat{k} + \hat{n} \cdot \hat{r}_0 \approx 2 \]

and with the third condition we obtain

\[ U(x_0, y_0, z_0) = \frac{i \exp(ikr_0)}{\lambda r_0} \int \int \exp \left[ -\frac{ik}{r_0} (xx_0 + yy_0) \right] dx dy \]

This equation gives the amplitude at \((x_0, y_0, z_0)\) in the far field when the aperture is illuminated by a plane wave at normal incidence.

Since a point of observation lies in a plane (the observation screen) located at a fixed distance \( z_0 \) from the aperture,

\[ r_0 = \sqrt{x_0^2 + y_0^2 + z_0^2} \]

\[ = z_0 \left( 1 + \frac{x_0^2 + y_0^2}{z_0^2} \right)^{\frac{1}{2}} \]

\[ \approx z_0 + \frac{x_0^2 + y_0^2}{2z_0} \]
A.4 Fresnel Diffraction

Using this expression for $r_0$ in the exponent $\exp(ikr_0)$ but using $r_0 \simeq z_0$ elsewhere, we have

$$U(x_0, y_0) = \frac{i \exp(ikz_0)}{\lambda z_0} \exp\left(\frac{tk}{2z_0}\right) \iint_S \exp\left(-\frac{tk}{z_0}(xz_0 + yy_0)\right) dx dy$$

Finally, let the aperture be filled with an arbitrary distribution $f(x, y)$ that is zero outside $S$, then

$$U(x_0, y_0) = \frac{i \exp(ikz_0)}{\lambda z_0} \exp\left(\frac{tk}{2z_0}\right) \int_{-\infty}^{\infty} f(x, y) \exp\left(-\frac{tk}{z_0}(xz_0 + yy_0)\right) dx dy$$

where $U$ is the amplitude of the Fraunhofer diffraction pattern produced by the aperture amplitude $f$ at a distance $z_0$ from the aperture. $f$ can be taken to describe the 'shape' of the aperture.

A.4 Fresnel Diffraction

Consider the Kirchhoff diffraction integral derived earlier

$$U(r_0) = \frac{i k \alpha}{4\pi r_0} \iint_S \exp(ik \cdot r) \exp(ik |r - r_0|) dS$$

where

$$\alpha = \hat{n} \cdot \hat{k} + \hat{n} \cdot \hat{r}_0$$

The Fresnel approximation considers only the field which occupy a small angle to the optical axis (z axis), in the exponential function where small changes in $r$ result in a large phase changes. We consider the expansion of $|r - r_0|$ to second order and retain the term $r^2/2r_0$, i.e.

$$|r - r_0| = r_0 - r \cdot r_0 + \frac{r^2}{2r_0} + \cdots$$

$$\simeq r_0 - r \cdot r_0 + \frac{r^2}{2r_0}$$
A.4 Fresnel Diffraction

This approximation is necessary when the diffraction pattern is observed in what is called the intermediate field or Fresnel zone.

\[ U(r_0) \approx \frac{ik\alpha}{4\pi r_0} \exp(ikr_0) \int \int_S \exp(ik \cdot r) \exp(-ikr_0 \cdot r) \exp \left( \frac{ikr^2}{2r_0} \right) dS \]

Consider a Cartesian coordinate system where

\[ r = \hat{x}x + \hat{y}y + \hat{z}z \]

\[ r_0 = \hat{x}x_0 + \hat{y}y_0 + \hat{z}z_0 \]

Here, \( x_0 \) and \( y_0 \) are taken to represent the position on a flat screen at a fixed distance \( z_0 \) from the aperture. Further, let

- \( \hat{k} \approx \hat{z} \) plane wave at normal incidence to the aperture
- \( r_0 \approx \hat{z} \) observations at small angles only
- \( kz \rightarrow 0 \) 'infinitely thin' aperture

Then,

\[ U(x_0, y_0) = \frac{i}{\lambda} \exp(ikr_0) \int \int_S \exp \left[ -\frac{ik}{r_0} (xx_0 + yy_0) \right] \exp \left[ \frac{ik}{2r_0} (x^2 + y^2) \right] dx dy \]

As with the analysis associated with Fraunhofer diffraction, we substitute

\[ r_0 \approx z_0 + \frac{x_0^2 + y_0^2}{2z_0} \]

into the exponent \( \exp(ikr_0) \) but use \( r_0 \approx z_0 \) elsewhere. By introducing an aperture amplitude function \( f(x, y) \), the amplitude function is then given by

\[ U(x_0, y_0) = \frac{i}{\lambda} \exp(ikz_0) \exp \left( \frac{ik}{2z_0} (x^2 + y^2) \right) \int \int f(x, y) \exp \left[ -\frac{ik}{z_0} (xx_0 + yy_0) \right] \exp \left[ \frac{ik}{2z_0} (x^2 + y^2) \right] dx dy \]

From the last equation we see that \( U \) is essentially (ignoring scaling constants) given by the convolution of

\[ f(x, y) \text{ with } \exp \left( \frac{ik}{2z_0} (x^2 + y^2) \right) \]
or

\[ U(x, y) = \frac{i \exp(ikz_0)}{\lambda z_0} f(x, y) \otimes \exp \left( \frac{ik}{2z_0} [x^2 + y^2] \right) \]

where \( \otimes \) denotes the 2D convolution integral.

Ignoring scaling, we can define the Fresnel transform as

\[ s(x, y) = \exp[\alpha(x^2 + y^2)] \otimes f(x, y) \]

where \( \alpha = k/2z_0 \).

This result describes Fresnel diffraction - the diffraction pattern observed in the intermediate field.
Appendix B

Research Software

During the development of this research thesis, a software package was developed, written using MATLAB 6.5 Release 13. The package includes all functions used during the research, supported with a prototype graphical user interface. The package is designed to be a research environment rather than an end-user package and as such, should be of a value to a future researcher in this area. The MATLAB code is provided in the accompanying CD given at the back of this thesis.

The main interface for this package is shown in Figure B.1.

B.1 Running The Package

The accompanying CD contains all the MATLAB files (*.fig; *.m) needed to run the software. The following steps are used to run the package:

(i) Copy Mat_Programs folder from the CD to your hard disk

(ii) Start MATLAB

(iii) Add CommonTools folder to the execution path of MATLAB.
(a) From the MATLAB File menu choose "Set Path".
(b) Click on "Add with sub folders" button. In the window that appears, select the Mat_Programs\CommonTools folder.
(c) Click on "Save" and then on "Close" buttons.
(iv) From the MATLAB File menu, choose "Open". In the window that appears, select the Mat_Programs folder.
(v) Double-click on "Holography1.m" file.
(vi) To execute the file press "F5".

B.2 User Interface

The main window is composed of a menu bar and six frames. Each frame can hold an image. The main menu and its sub-menus are shown in Figure B.2. All operations applied on each image are grouped in a context menu that can be activated by the right-click option button beneath each frame. The context menu and its sub-menus are shown in Figure B 7.

B.2.1 Main Menu

File SubMenu

File menu provides several important commands which are as follows:

File>Load
This package supports only images in Bitmap format, and six images can be opened at any one time. To open an image, choose this option. Then, select the colour mode. Finally, select the file from the "Open File" dialog box. The package provides two colour mode: grey and RGB. Grey scale images are made up of 8
Figure B.2. Main menu

- Coding
  - File
  - Operation
  - On Two Images
  - Experiments
  - Help

Load
- Generate Diffuse Operator
- Scan
- Print All
- Reset All
- Exit

Diffusion
- Add
- Subtract
- Multiply

DeDiffusion
- XOR
- Metrics

Gray
- RGB

About DocMarking
bits of information per pixel and use 256 shades of grey to simulate gradations in colour. RGB images use three colours (red, green, blue) to produce up to 16.7 million colours on-screen. RGB images are three-channel images, so they contain $24(8 \times 3)$ bits per pixel. As requested, the package will convert from grey-scale to RGB images and vice versa. The images are displayed in the first available frame in the main window (i.e. starts from the first frame, progresses toward the end and starts again). The size of each image ($height \times width$) is displayed as a part of the image caption.

File>Generate Diffuse Operator

This option provides the ability to generate the diffusion operator used to diffuse the watermark. Figures 4 5, and B.3 shows the interface used to generate a grey-scale and RGB noise field respectively. The interface provides different operators such as Gaussian, Uniform, Fractal, Chaos, etc. Figure 4.4 shows the menu that lists all these operators. Each operator has its own parameter set. For RGB noise, it is necessary to specify the operator and its parameter set for each channel. Current (active) channel(s) are those that have their checkbox checked. To regularize the spectrum of the noise, click on “Clear Amplitude” checkbox. To replace all zeros in the noise’s amplitude spectrum by one, click on “Clear Zeros” checkbox (for more details refer to Adaptive filter: Option 2).

File>Scan

Scanned images can be imported directly from any scanner that supports the TWAIN interface. Before scanning an image, it is necessary to make sure that the software for ‘driving’ the scanner is installed. To import the scanned image, choose the scan option and then choose the desired colour mode. The scanner’s interface will appear directly. When the scanning is finished, the image appears in the next available frame in the main window.

File>Print All

Clicking this option brings up the “Print” dialog box. This option prints the
Figure B.3: RGB noise interface

Note: any change will be in effect after clicking on the above.
B.2 User Interface

Figure B.4: 'Select Two Images' interface.

main window Figure B.1 is a sample printout.

File>Reset All
Using this option, all frames are cleared. This option is useful for initiating a new session.

File>Exit
Using this option terminates the application.

"Operations on Two Images" SubMenu

In this menu, all binary operations (i.e. as applied to two input images) are listed. The two input images must be loaded and displayed in any frame before the operation can be applied. Whenever executing any of these binary operations, a window is generated to locate the input and the output images. This window is shown in Figure B.4.

Diffusion
Clicking on this option displays the diffusion interface (see Figure 4.6). It shows the diffusion operator in the left frame and the diffused image in the right frame. By
B.2 User Interface

clicking on "Change The Diffuse Operator" button, the diffusion operator can be changed. To see the extracted watermark directly, click on "De-Defusion" button. The interface provides three ways to extend the small image (the two input images must have the same size) which are explained later. By default, this interface convolves the first image with the second image. When the "Apply2" button is pressed, the output is divided by the power spectrum of the noise. To go back to the default output, click on "Apply1" button (for more details refer to Adaptive filter: option 1).

De-Diffusion

Clicking on this option displays the de-diffusion interface (see Figure 4.8). The interface shows the extracted watermark. By default, this interface uses the matched filter. To use the inverse filter or the Weiner filter (i.e., Adaptive filter: Options 4 and 5) the corresponding sliders can be used which are set between the frequency components used in matched filter and those used in inverse or Weiner filter. Activating the corresponding "Apply" button produces the result. The maximum and minimum energy (i.e., power spectrum) is shown in the top left corner of the interface. The current value of the slider is shown in the same corner. The two buttons: "Set Current as a Slider Max" and "Set Current as a Slider Min" are used to change the slider's range which can be used to set the upper and lower bounds. This interface (like the diffusion interface) provides three ways to extend the small image.

Add

In normal addition, the two input images \((f \text{ and } s)\) are added to each other using linear addition. The linear addition is \(r = f + \alpha s\), where \(\alpha\) is the embedding strength (or SNR). The resultant image is shown in the interface. The default value for \(\alpha\) is 100% which can be changed using the existing slider. The overlay option uses the following formula: \(r = (1 - \alpha)f + \alpha s\). The interface is shown in Figure B.5.
Subtract
This option subtracts two images from each other using, i.e. $r = f - \alpha s$. The interface used here is the same as the “Add two Images” interface.

Multiplication
This option is suitable for adding a texture to a text image. Here, the addition is a normal addition except that the texture is added only to the background. The interface used here is the same as the “Add two Images” interface. For more detail on this option refer to Section 5.1.2.

XOR
Using this option, the two input images will be XORed to each other.

Metrics
By using this option, one can measure the distance between two images using different metrics. The differences are calculated in the spatial domain or the frequency domain together with the histogram statistics. These metrics are shown
B.2 User Interface

in Figure B.6.

Experiments Menu

This menu contains only one option, named, Geometric. This option generates the interface used to test the effect of geometric attacks on the system. The interface is shown in Figure 6.17. The input images are the diffused watermark and the diffusion operator. In this interface, one can test the effect of rotation, cropping, scaling and translation. The extracted watermark is shown directly in the right frame while the attacked image is shown in the left frame. Any attack can be reversed (e.g. crop an image 20 pixels from the top, then -20 pixels from the top) until the corresponding “Apply” button is executed. This is helpful in monitoring or testing the effect of combined attacks. The parameters of each attack are described later.

B.2.2 Context Menu

This menu is composed of all the unary operations. Figure B 7 shows all of the operations available. To activate this menu, right-click the radio button beneath the desired image frame. The output can be displayed in the same frame or in a separate window.

Set Next

This option is used to set the current frame as the ‘next available frame’. The output of the next operation (from the file menu or a binary operation) is then displayed in this frame; the next output is displayed in the next frame and so on.

Un-Dock

This option is used to show the current frame in a separate window.

Save
## B.2 User Interface

Figure B.6: 'Metrics' interface.
Figure B.7: Context menu.
B.2 User Interface

The save option saves the current image using a BMP format. The option brings up the ‘Save as’ dialog box.

Print
The print option prints the current image.

Frequency(Mesh) and Frequency(Image)
These two options display the amplitude, real, imaginary and the phase spectrum using a 3D mesh or 2D image.

Histogram
The histogram shows how the pixels values are distributed in an image. If the current image is RGB, then the Figure shows the histogram for each channel.

Channel
This option is useful for RGB images. It displays the contents of each channel.

Histogram Equalization
This option applies ‘histogram equalization’ on the current image. Histogram equalization enhances the contrast by spreading the histogram of the input image so that it will span a fuller range of the grey scale.

Scale(0..1)
The output image contains values in the range 0.0 (black) to 1.0 (white).

Amplitude Change
The changes in amplitude (contrast changing) is represented by $c = vc$ where $c$ is the original image and $v$ is a scaling factor. The scaling factor can be changed using the corresponding slider.

Inverse
The inverse command inverts an image. When you invert an image, the brightness value of each pixel in the channels is converted to the inverse value on the 256-step colour-value scale. For example, a pixel with a value of 255 is changed to 0, and a pixel with a value of 5 to 250.
B.2 User Interface

Padding techniques: (a) Value padding, (b) Wrapping, (c) Border padding

Padding: Value Padding, Wrapping and Border Padding
All binary operations require two equal-size images. If one image is smaller than
the other, then padding is used to solve this problem. Padding can be done in
three ways:

(i) Value Padding - pads the image (in both dimensions) with a specific value.

(ii) Wrapping - assumes that the image is wrapped back on itself in both dimen-
sions.

(iii) Border Padding - assumes that the pixels that lie beyond the ends of the
image dimensions take on the value of the end points in each dimension.
This method is one of the most widely used of its type.

These approaches are illustrated in Figure B.8.

Threshold
This option is used to convert grey-scale or colour images to high-contrast, black-
and-white images. The interface (see Figure 6.8) lets you specify a certain level
as a threshold. The threshold can be changed by the corresponding slider and the
result shown directly.

Set Pixel
This option is used to assign a new value to a specific pixel.

Bar Coding
This sub menu is the implementation for the ideas presented in Section 7.2:
B.2 User Interface

(i) Uniform Binary Modulation - this option XOR’s the current image with uniform noise. The same option is used for coding and decoding.

(ii) 2D Replacement (XOR) - Figure B.9 and B.10 show the interface that appears when the user clicks on “2D Barcode Modulation” and “2D Barcode Demodulation” respectively. In coding, load the desired 2D barcode (right image) first. Then, generate the key using the XOR operation (left image). The key image must be saved in order to be used in decoding phase. The save option saves the key image in ‘tif’ format (lossless compressed format). In decoding, load the key first (right image), then generate the original image using the XOR.

(iii) 2D Replacement (Convolution) - the same as above, except that convolution and correlation are used instead of XOR (left image).

JPEG
This option applies JPEG lossy compression to the current image. A scale between 0 and 100 provides the bounded with this kind of compression. This scale can be
changed using the associated slider.

Filters: Remove Foreground
This option is used to remove the foreground (i.e., text) using the algorithm discussed in Section 5.1.4. The input parameters are shown in Figure 5.8.

Filters: Gaussian LPF, Ideal LPF and Ideal HPF
These options apply the Gaussian Low-Pass Filter, Ideal Low-Pass Filter and Ideal High-Pass Filter respectively. The width of the Gaussian and the cut-off frequency must be provided to the Gaussian and ideal filters respectively.

Geometric Attack
The available geometric operations are

(i) Scale Up and Down: Resizes the current image using the “bicubic” interpolation method.

(ii) Translation: Zero-Padding the current image in any direction (top, bottom, left and right). An option is added to resize the image to its original size after translation.

(iii) Crop: Discards the area outside a rectangular selection and keeps the same
resolution as the original. To specify the cropping rectangle, enter the coordinates of the upper left corner, the width, and the height of the rectangle.

(iv) Rotate: The rotate option rotates the current image by a specified angle. The rotation is in a counter-clockwise direction, executed using the "bicubic" interpolation method. The output image is larger than the original. To include only the central portion of the rotated image and utilize the same size as the original image, click on the corresponding option box.

Finally, Figure B.11 shows the "About" window. A diffused watermark can be added to the window in order to protect the copyright of this software.
References


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


