2006

Improved content based watermarking for images

Arvind Kumar Parthasarathy
Louisiana State University and Agricultural and Mechanical College, aparth1@lsu.edu

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_theses

Part of the Electrical and Computer Engineering Commons

Recommended Citation
Parthasarathy, Arvind Kumar, "Improved content based watermarking for images" (2006). LSU Master's Theses. 3555.
https://digitalcommons.lsu.edu/gradschool_theses/3555

This Thesis is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Master's Theses by an authorized graduate school editor of LSU Digital Commons. For more information, please contact gradetd@lsu.edu.
IMPROVED CONTENT BASED WATERMARKING FOR IMAGES

A Thesis

Submitted to the Graduate Faculty of the Louisiana State University and Agricultural and Mechanical College in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering

in

The Department of Electrical and Computer Engineering

By

Arvind Kumar Parthasarathy
Bachelor of Engineering, Sri Chandrasekharendra Saraswathi Viswa Maha Vidyalaya, 2003 Kanchipuram, India
August 2006
To my mother

Chitra Parthasarathy
Acknowledgements

This work would not have been possible without the constant encouragement and support I received from Dr. Subhash Kak, my advisor and mentor. I would like to express my sincere thanks to him. I would also like to offer my gratitude to Dr. Xue-Bin Liang and Dr. Hsiao-Chun Wu for serving on my thesis advisory committee.

I would like to extend my sincere thanks to Dr. Chacko John for his support and encouragement.

I would like to thank my good friends Chetan Salimath and Srinivas Gopal Krishna for all their help and support. I would also like to thank all my friends who have helped me all through my stay at LSU and have made it a memorable one.

Last, but by no means the least, I express my grateful appreciation to my parents for their love, encouragement and support. Through them I have learned that perseverance is the key to success.
Table of Contents

Acknowledgements........................................................................................................... iii

List of Tables........................................................................................................................ vi

List of Figures........................................................................................................................ vii

Abstract................................................................................................................................. x

Chapter 1 Introduction ........................................................................................................ 1
  1.1 Background.................................................................................................................. 2
  1.2 Steganography ............................................................................................................ 3
  1.3 Digital Watermarking................................................................................................. 3
  1.4 Applications of a Watermark .................................................................................... 5
  1.5 Types of Watermark ................................................................................................ 5
  1.6 Outline of Thesis ...................................................................................................... 6

Chapter 2 Review of Literature ......................................................................................... 8
  2.1 Digital Watermarking Techniques ........................................................................... 8
  2.2 Spatial Watermarking ............................................................................................... 9
  2.3 Transformed Domain Watermarking ...................................................................... 11
      2.3.1 Frequency-based Watermarking ...................................................................... 11
      2.3.2 Singular Value Decomposition ....................................................................... 12
  2.4 Correlation-based Watermarking ............................................................................ 14
  2.5 CDMA Spread Spectrum Watermarking ................................................................. 15
  2.6 Watermarking Attacks ............................................................................................. 16

Chapter 3 Content Based Image Watermarking ............................................................. 17
  3.1 Pseudorandom Noise Sequences (PN Sequences) .................................................. 18
      3.1.1 Properties of PN Sequences ........................................................................... 20
  3.2 Decimal Sequences (d-sequences) .......................................................................... 22
      3.2.1 Properties of d-sequences ............................................................................. 22
          3.2.1.1 Cross-correlation Property ................................................................... 25
          3.2.1.2 Autocorrelation Property ................................................................... 25
      3.2.2 Generation of d-sequences ............................................................................. 26
  3.3 Proposed Scheme ..................................................................................................... 28
      3.3.1 Just Noticeable Distortion (JND) Visual Mask ............................................... 28
      3.3.2 Watermark Embedding .................................................................................. 39
      3.3.3 Watermark Detection .................................................................................... 43
      3.3.4 Watermark Evaluation ................................................................................... 44

Chapter 4 Experimental Results ....................................................................................... 46
  4.1 Attacks and Analysis of Results .............................................................................. 59

Chapter 5 Conclusions ....................................................................................................... 66
List of Tables

Table 3.1  DCT values for a $8 \times 8$ block ................................................................. 31
Table 4.1  WPSNR table for different primes ............................................................... 52
Table 4.2  WPSNR table for different scaling factors ................................................. 54
List of Figures

Figure 1.1 Watermark Embedding ......................................................... 4
Figure 1.2 Watermark Extraction .......................................................... 4
Figure 2.1 Watermarking Schemes ......................................................... 10
Figure 3.1 PN Generator ................................................................. 19
Figure 3.2 Output Table for PN Generator .............................................. 19
Figure 3.3 Decimal Sequence Generator ................................................. 27
Figure 3.4 Definition of DC/AC components for a $8 \times 8$ DCT block .......... 31
Figure 3.5 Edge Extraction using Canny Operator .................................... 34
Figure 3.6 Edge Extraction based on Phase Congruency ............................ 35
Figure 3.7 Corner Detection using Curvature Scale Space Method ................. 37
Figure 3.8 Definition of DCT regions .................................................... 41
Figure 4.1 Lena Reference Image ......................................................... 46
Figure 4.2 Watermarked Image .......................................................... 47
Figure 4.3 Original Watermark .......................................................... 47
Figure 4.4 Recovered Watermark ........................................................ 47
Figure 4.5 Sufficient Statistic for $q = 2467$ ............................................. 47
Figure 4.6 Watermarked Image .......................................................... 48
Figure 4.7 Embedded Watermark ........................................................ 49
Figure 4.8 Recovered Watermark ........................................................ 49
Figure 4.9 Sufficient Statistic for $q = 8069$ ............................................. 49
Figure 4.10 Watermarked Image ........................................................ 50
Figure 4.11 Embedded Watermark ........................................................ 50
Figure 4.12  Recovered Watermark ................................................................. 50
Figure 4.13  Watermarked Image ................................................................. 51
Figure 4.14  Embedded Watermark ............................................................... 51
Figure 4.15  Recovered Watermark ............................................................... 51
Figure 4.16  Watermark ............................................................................. 52
Figure 4.17  Watermark ............................................................................. 52
Figure 4.18  Watermark ............................................................................. 53
Figure 4.19  Watermark ............................................................................. 53
Figure 4.20  WPSNR vs. Scaling Factor Plot for Lena and Boat ................... 53
Figure 4.21  Normalized Texture Plot for Lena ............................................ 55
Figure 4.22  Normalized Edge plot for Lena ................................................. 56
Figure 4.23  Normalized Corner Plot for Lena ............................................. 56
Figure 4.24  Normalized Luminance Plot for Lena ....................................... 57
Figure 4.25  Normalized JND values for Lena .............................................. 57
Figure 4.26  Normalized JND values for Boat .............................................. 58
Figure 4.27  JPEG Compression ($q=45$) ...................................................... 60
Figure 4.28  Recovered Watermark ............................................................... 60
Figure 4.29  JPEG Compression ($q=40$) ...................................................... 60
Figure 4.30  Recovered Watermark ............................................................... 60
Figure 4.31  JPEG Compression ($q=35$) ...................................................... 61
Figure 4.32  Recovered Watermark ............................................................... 61
Figure 4.33  JPEG Compression ($q=30$) ...................................................... 61
Figure 4.34  Recovered Watermark ............................................................... 61
Figure 4.35  Uniform Gaussian Noise ........................................................... 63
Figure 4.36  Recovered Watermark…………………………………………………………..63
Figure 4.37  Salt 'n' Pepper Noise ……………………………………………………….. 63
Figure 4.38  Recovered Watermark…………………………………………………………..63
Figure 4.39  Median Filter ………………………………………………………………….. 64
Figure 4.40  Recovered Watermark…………………………………………………………..64
Figure 4.41  Contrast Sharpening Filter ………………………………………………….. 65
Figure 4.42  Recovered Watermark…………………………………………………………..65
Abstract

Due to improvements in imaging technologies and the ease with which digital content can be created and manipulated, there is need for the copyright protection of digital content. It is also essential to have techniques for authentication of the content as well as the owner. To this end, this thesis proposes a robust and transparent scheme of watermarking that exploits the human visual systems’ sensitivity to frequency, along with local image characteristics obtained from the spatial domain, improving upon the content based image watermarking scheme of Kay and Izquierdo [1]. We implement changes in this algorithm without much distortion to the image, while making it possible to extract the watermark by use of correlation. The underlying idea is generating a visual mask based on the human visual systems’ perception of image content. This mask is used to embed a decimal sequence, while keeping its amplitude below the distortion sensitivity of the image pixel.

We consider texture, luminance, corner and the edge information in the image to generate a mask that makes the addition of the watermark less perceptible to the human eye. The operation of embedding and extraction of the watermark is done in the frequency domain thereby providing robustness against common frequency-based attacks including image compression and filtering. We use decimal sequences for watermarking instead of pseudo random sequences, providing us with a greater flexibility in the choice of sequence. Weighted Peak Signal to Noise Ratio is used to evaluate the perceptual change between the original and the watermarked image.
Chapter 1

Introduction

The use of Internet as a platform for distribution of digital media has grown in recent years causing serious concerns about unauthorized use and manipulation of this digital content. Enforcement of intellectual property rights has become an important issue since copying and editing of digital information is very easy [2]. Multimedia content such as pictures, video files and audio files are prone to being accidentally or maliciously modified, altered or destroyed. To deal with the issue of data integrity, authentication techniques are required to verify accuracy, correctness and validity of the digital content. Digital Rights Management (DRM) is an approach that addresses the issue of piracy and ensures that the digital media is distributed to authorized parties only.

The traditional forms of protecting information are steganography and cryptography, which are both used in the implementations of DRM. Steganography is a relatively insecure way of hiding information, but it is the precursor to a more robust method called watermarking, in which the message is hidden so that the source can be tracked or verified.

Cryptography is commonly used for authenticating the integrity of digital data, and it is achieved using digital signatures [3]. The traditional methods of data authentication used security and encryption to resolve the issue of digital rights. This has slowly given way to the second generation DRM, which is broader in scope, covering not only the above mentioned issues but also the description, identification, trading, protection, monitoring and tracking of all forms of rights usage [4].
1.1 Background

Digital watermarking, which is hiding useful information pertaining to the owner or the content creator, was initially used in paper and money as a measure of genuineness. Although as a means of authentication of digital content may not prevent unauthorized use, it certainly is a mechanism to track the owner of the digital content. DRM techniques are being researched to establish more secure protocols to safeguard the information as well as control their distribution.

There are various implementations of DRM and content management systems. Serial Copy Management Systems (SCMS) was created to prevent anyone from making second generation or serial copies of the original but it does not limit the number of copies made from the master file and Content Protection for Recordable Media and Pre-Recorded Media (CPRM/CPPM), which is used to control the copying, addition and deletion of digital media on a host device like personal computers.

Another application of DRM is the Content Scrambling System (CSS), a DVD encryption scheme where key sets are licensed to manufacturers who incorporate them into DVD players. The DVD players were equipped with the CSS Decryption module and on providing the proper key the video is decrypted and played. But this algorithm was easily susceptible to a brute force attack. The presence of an ideal watermark, which would persist, despite encryption, seems more viable for enhancing the already existing DRM architectures. One of the widely used applications is the use of DRM protected songs that can only play on a specific hardware device.
1.2 Steganography

Steganography means “covered writing” in Greek and it can be described as the science of writing hidden messages such that no one other than the intended party is aware of the existence of the message. This is different from cryptography where the contents or the meaning of the content is secured. Historically, many techniques of steganography have been employed including the use of wax covered tablets and secret inks. Steganography can be used for digital watermarking, where a message is hidden in an image often called the cover object, so that it can be used to trace and verify its original owner. An ideal steganographic system is able to conceal a large amount of information by ensuring that the changes made to the cover object due to the injection of the hidden message is largely indistinguishable or visually negligible. On the other hand, an ideal cryptographic system would embed information that is robust to all forms of attacks without making the cover object completely unusable.

1.3 Digital Watermarking

Watermarking consists of three main stages: insertion, detection and the removal of a watermark. The detection and removal are usually considered together. A generic block diagram is shown in the schematic Figure 1.1 and Figure 1.2. Each of the stages involves the use of a secret user key. An embedding algorithm is used to embed the watermark in the image. The extraction algorithm recovers the watermark, which requires the same secret key that was used for watermark embedding. Extraction of a watermark can be separated into two phases, namely, watermark detection and watermark recovery.
Figure 1.1 Watermark Embedding

Figure 1.2 Watermark Extraction
The extraction is carried out using the original document (non-blind watermarking scheme) or in its absence (blind watermarking scheme). Image watermarking depends largely on the domain in which the watermarking is done.

1.4 Applications of a Watermark

There are several applications of watermarking. Some of the important ones are enlisted below:

- **Copyright Protection/User Authentication**: Digital content is watermarked identifying the copyright ownership.

- **Piracy Protection**: Digital content is watermarked and only a device capable of replication of this content can detect such a watermark. This is to basically prevent the unauthorized replication of content.

- **Broadcast Monitoring and Tracking**: Digital watermarks are used to track a broadcast of media over a channel for various purposes. Time stamps are integrated in the watermarks to identify the owner as well as the time the digital content was created. Watermarks are also be used to track the users who replicate the content illegally.

- **Medical Applications**: Watermarking is used to identify the medical x-ray images and other records of patients thereby reducing the chances of tampering of the medical records.

1.5 Types of Watermark

Based on the type of application, watermarks are classified in a number of ways. Some types of watermarking schemes are explained below:
• Visible and Invisible watermarks: The user of the content can see a Visible watermark. The watermark usually identifies the owner of the content and consists of a logo or seal of an organization. This watermark is visible although it does not totally obscure the primary image. In this case it is important to overlay the watermark in such a way so as to make it robust to attacks. On the other hand, invisible watermarks are imperceptible and they can barely be detected.

• Robust and Fragile watermarking: Robust watermarks are those that are difficult to remove from digital content. In other words intentional or incidental distortions like compression, scaling, cropping, filtering etc to a watermarked image does not affect the watermark. Fragile watermarks are those that are easily destroyed by tampering or modifying the watermarked content. Hence the absence of a watermark to a previously watermarked content points to the conclusion that the data has been tempered with.

• Asymmetric and Symmetric watermarking: In asymmetric watermarking different keys are used for embedding and detecting a watermark while symmetric watermarking uses the same set of keys for both operations.

1.6 Outline of Thesis

The main idea of this thesis is to improve upon the content based image watermarking scheme of Kay and Izquierdo [1] by implementing changes in the algorithm that embeds the watermark information, without much distortion to the image, while making it possible to extract the watermark by use of correlation. We begin by generating a mask based on the Human Visual systems’ perception of image content, which is used to embed a watermark such that its amplitude is kept below the distortion
sensitivity of the pixel. Texture, luminance, corner and the edge information in the image goes into the mask, which makes the addition of the watermark less perceptible to the human eye. The operation of embedding and extraction of the watermark is done in the frequency domain thereby providing robustness against common frequency-based attacks including image compression and filtering. We use decimal sequences for watermarking instead of pseudo random sequences enabling an easier generation and a much better detection, of the watermark.

The thesis is organized as follows: In chapter 2 we perform a survey of the basic watermarking schemes. Chapter 3 deals with the properties of decimal sequences and pseudo random sequences as well as their generation and the explanation of the proposed scheme in detail. Chapter 4 deals with the results and further analysis of the results of the proposed watermarking scheme. Finally, chapter 5 presents the conclusions.


Chapter 2

Review of Literature

This chapter discusses some of the watermarking techniques that have been implemented so far. We will also discuss briefly the results, advantages including security, robustness, capacity, complexity and the limitations of these schemes. Finally, various types of attacks on watermarking schemes are discussed.

2.1 Digital Watermarking Techniques

Watermarking takes place either in the spatial domain, where the addition of the watermark is done directly to the pixel values of the image, or in a transformed domain like discrete cosine transformed domain (DCT) or the discrete wavelet transformed domain (DWT). The watermarking in a transformed domain provides more robustness to many forms of attack.

The basic requirement of a watermarking scheme is a watermark embedding system and a watermark extraction system. The input to a watermark embedding system is either a randomly generated sequence of bits or the watermark itself and the output is the watermarked image. The watermark extraction system is used to determine whether or not a watermark has been added. A watermarking system should be designed considering the following factors:

• Imperceptibility of the watermark: This is perhaps the most important property of all watermarking schemes. The watermark must be embedded in the image in such a way that the resulting watermarked image is not visually distorted. This property is related to the robustness of the watermark and hence an optimal
balance between imperceptibility and robustness must be achieved by the watermarking scheme.

- Robustness of the watermark: This is another important property of a good watermarking scheme. Most schemes require that the watermark be recovered even if the watermarked image is distorted and this distortion may be accidental or deliberate. There are many types of attacks for a watermarking scheme including noise introduction, filtering, image compression, cropping, re-sizing, etc and the robustness to such attacks depends on the type of watermarking. In other words a fragile watermarking scheme does not require being robust against any attacks on the image as it is designed to confirm the occurrence of an attack.

- Payload: It is described as the total amount of watermark information bits that can be stored in the image. It is generally considered that the more the watermark bits (payload) in an image the higher the robustness of the watermark.

- Efficiency: It is the speed with which an algorithm performs the insertion and the detection of a watermark.

The block diagram for various watermarking schemes that have been implemented so far is shown in Figure 2.1.

### 2.2 Spatial Watermarking

A simple and widely used method of inserting a watermark is the Least Significant Bit (LSB) watermarking. In this method, the lower order bit of the selected pixel in the image is used to store the watermark information bit, which is an example of spatial watermarking. The amount of information that can be stored by this method is
limited therefore it is highly susceptible to attacks. All that the attacker has to do is to remove the LSB in the watermarked image to get rid of the watermark.

Figure 2.1 Watermarking Schemes

Another modification to this method is to use a secret key to choose a random set of LSBs and replace them with the watermark bits. One way of selecting this element is
by using a pseudo random number generator [5]. This type of watermarking is invisible as only one bit is being changed. But most image manipulations such as cropping, re-sampling or format conversions will result in the watermark information being lost. Hence this scheme, although being highly imperceptible, is not very robust.

2.3 Transformed Domain Watermarking

2.3.1 Frequency-based Watermarking

To obtain better imperceptibility as well as robustness, the addition of the watermark is done in a transformed domain [6], [7], [8], [9]. DCT and DWT are two such popular transforms, operating in the frequency domain. Frequency-based techniques are very robust against attacks involving image compression and filtering because the watermark is actually spread through out the image, not just operating on an individual pixel. This is just one of the many advantages of embedding the watermark in a transformed domain as opposed to watermarking in the spatial domain. It is also well known as to how efficiently the transformed coefficients can be altered in order to minimize perceptual/visual distortion in the watermarked image [2] which explains why such schemes are widely implemented.

The LSB method is also applied in the frequency domain by selecting the pixel based on the frequency. Most methods involve the modification of transformed coefficients based on their frequencies due to its imperceptibility. In a frequency-based watermarking scheme the watermark, upon inverse transformation to the spatial domain, is dispersed throughout the image making it very difficult for an attacker to remove the watermark without causing significant damage to the image. The watermark signal that is typically used in image watermarking schemes is a pseudorandom signal with low
amplitude. Watermarking in the DCT domain is usually performed on the lower or the mid-band frequencies, as higher frequencies are lost when the image is compressed. DCT watermarking can be done for an entire image taken together or block-wise. In both these methods the image is transformed into its DCT coefficients, which arranges them based on the frequency, and the watermark is added to these coefficients. Finally the watermarked coefficients are inverse-transformed into the spatial domain thereby spreading the watermark throughout the image or the block of the image.

In the frequency-based watermarking technique based on DWT, the watermark is added to the low and the high frequency values of the DWT coefficients. In some schemes only the LL band is modified while in others the watermark is added to all the bands. In almost all the transformed domain watermarking techniques there is a trade-off between robustness and imperceptibility. The watermark, if embedded in the perceptually significant components, would result in a visible change in the final watermarked image. On the other hand, if it were embedded in the perceptually insignificant components then they would not be as robust and hence less resilient to most attacks.

2.3.2 Singular Value Decomposition (SVD)

There are various domains used to transform an image and in the singular value decomposition (SVD) method the singular values are modified. The idea of using singular values for watermarking was explored a few years ago by Liu and Tan [10]. Any matrix $A \in \mathbb{R}$ of size $(m \times n)$ can be decomposed into 3 matrices,

$$A = U \Sigma V^T$$

where $U$ and $V$ are orthogonal matrices,

$$UU^T = I, \text{ and } VV^T = I$$

where $\Sigma = \text{diagonal } \{\sigma_i \geq 0, i = 1, 2 \ldots n\} \in \mathbb{R}$ of size $(m \times n)$, are the singular values of $A$ or the square roots of the eigen values of $AA^T$ and $A^TA$. 
The columns of $U$ are called the left singular vectors of $A$ or the eigenvectors of $A A^T$ and the columns of $V$ are called the right singular vectors of $A$ or the eigenvectors of $A^T A$. The singular values specify the luminance of an image layer while the corresponding singular vectors specify the geometry of the image [11]. There are many approaches to SVD watermarking. One of them is to apply SVD to the whole image and modify all the singular values, i.e., add the watermark to the all the singular values. Embedding a watermark in the SVD domain results in very little perceptual difference and the largest of singular values changes very little for most common attacks.

Other techniques of SVD watermarking include performing SVD in the frequency domain, that is performing the SVD on the DCT coefficients [11] or the DWT coefficients [12] of the image and then adding the watermark to the singular values. The addition of the watermark to the SVD coefficients in the DCT domain may be done in different ways. Some suggest adding the watermark only to the mid frequency coefficients because of the conflict between robustness and transparency while others argue on adding to the low and high frequency coefficients of the AC component.

The results of SVD watermarking have been very encouraging because of the high capacity of embedding watermark information in the singular values and the minimal change in the quality of the final watermarked image. One limitation of this scheme is the requirement of the original image as well as the original watermark/secret keys in order to extract the watermark from the watermarked image, for it is a non-blind watermarking scheme. Two of the important applications of SVD watermarking are to disable unauthorized access to content and copyright protection.
2.4 Correlation-based Watermarking

As mentioned earlier, schemes are divided into blind, semi-blind and non-blind watermarking schemes based on requirement of the original image at the receiver. In most schemes, the watermark is typically a pseudo randomly generated noise sequence, which is detected at the receiver using correlation. In a blind watermarking scheme, the secret key, used to generate the pseudo random sequence is required to get back the watermark. The generalized algorithm of most correlation-based spread spectrum watermarking in a spatial domain is based on the following equation:

\[ WI(i, j) = I(i, j) + k \times W(i, j) \]

where \( WI \) = watermarked image,

\( I \) = original image,

\( k \) = scaling factor, and

\( W \) is the watermark.

The watermark \( W \) is a pseudo randomly generated noise, based on a secret key. The main requirement for a correlation-based algorithm is that the noise should be uniformly distributed and both the noise and the image content should not be correlated. At the receiver, the correlation is done between the noise generated using the same key and the possibly altered watermarked image. If this value is greater than a pre-determined threshold then the watermark is said to be present. There is a trade off between the imperceptibility of the watermark and the scaling factor because the greater the value of the scaling factor, the higher is the probability of not making an error in detection of the watermark. By choosing an optimum value for scaling factor, this method, although prone to errors, can to a large extent determine the presence of a watermark effectively.
2.5 CDMA Spread Spectrum Watermarking

Code Division Multiple Access (CDMA) is a form of spread spectrum where the signal i.e., watermark information, is transmitted on a bandwidth much greater than the frequency content of the original information, in this case, an image. In other words the transmitted signal bandwidth is much greater than the information bandwidth. Spread spectrum uses wide-band, noise-like signals, hence making it hard to detect. The band spread is accomplished by a pseudo-random code, which is independent of the data [13]. The code that appears to be random is actually deterministic.

Watermarking techniques that use spread spectrum communication principles insert watermarks over the spectrum in audio, video and image signals. Early methods implementing CDMA techniques are robust to noise and various compression schemes and the recovery of the watermark embedded in the image [7]. This technique is used to increase the payload of the watermark, thereby increasing the probability of the watermark being detected by a correlation-based technique. Spreading of the watermark throughout the message bits of the image ensures better security against unintentional or intentional attacks especially geometrical attacks including image compression as well as scaling.

The main drawback of CDMA in a spatial domain is the limited message capacity as compared to other correlation-based technique as the watermark recovery drops off for larger watermarks. Yet another disadvantage is the processing time, which increases exponentially with the size of the watermark [14].
2.6 Watermarking Attacks

Watermarking attacks are broadly classified as intentional or unintentional. The former is aimed at rendering a watermark unusable and testing the limitations of digital watermarking, which may, in turn, be used to improve the watermarking scheme. Unintentional attacks are usually not intended to attack a scheme but ultimately end up exposing the shortcomings of the watermarking method. We need to study these attacks to develop countermeasures that assist in protecting digital watermarking systems. Understanding the limitations of these methods will allow the development of more robust schemes that can survive manipulations and attacks. Some common attacks on watermarking schemes are mentioned below [15]:

- Geometric: Attacks that include cropping, scaling, rotation, and mosaicing
- Signal processing: Addition of noise, lossy compression like JPEG, dithering, denoising, non-linear and adaptive filtering, and superimposition
- Specialized attacks: Attacks that are based on the prior knowledge of method and this includes chrominance attack, and desynchronization attacks
Chapter 3

Content Based Image Watermarking

Much research has been done to increase the robustness and the data hiding capacity of watermarking techniques based on perceptual properties of the Human Visual System (HVS) [1], [2], [16], [17]. The development and improvement of accurate human vision models helps in the design and growth of perceptual masks that can be used to better hide the watermark information thereby increasing its security. There is a tradeoff between robustness and imperceptibility.

Most steganographic techniques that are designed to be robust must insert the watermark information into the cover image in a way that is perceptually significant. Other techniques that are relatively better at hiding information, like the LSB method, are highly vulnerable to having the embedded data distorted or quantized by lossy image compressions like JPEG. For obvious reasons, we will consider an invisible watermarking method that is capable of hiding the watermark information in the cover image in an unnoticeable way. This imperceptibility is obtained by considering the various properties of the HVS that make the scheme more robust to many types of attacks. Existing algorithms for watermarking still images usually work either in spatial domain or in transformed domain.

Our watermarking scheme deals with the extraction of the watermark information in the absence of the original image, i.e., blind watermarking. Hence we make use of correlation-based watermark detection. A decimal sequence is added, to the cover object, instead of a PN sequence, based on the actual watermark. The results and formulae are
based on a 512×512 size cover image and a block refers to a DCT block of size 8×8, which is used for better robustness against JPEG compression.

3.1 Pseudorandom Noise Sequences (PN Sequences)

A random sequence is defined as a sequence of random variables. A pseudorandom sequence is a finite sequence or string that is generated by an algorithm using an initial seed. These sequences appear to be random in nature, but in fact they can be determined provided we have sufficient knowledge of the algorithm being used for the generation and the initial state of the random number generator. There are certain properties of these sequences that make them very important in cryptography and encryption as well as in watermarking.

The generation of a random sequence is achieved in a way, that represents the outcome of tossing a fair unbiased coin where the occurrence of a head being denoted as +1 and a tail as -1. Such a sequence would be perfectly random as the probability of obtaining a head or a tail would be the same, equaling half.

PN sequences are generated using simple cascaded linear feedback shift registers (LSFR) as shown in Figure 3.1. A PN sequence is termed as a maximal length PN sequence if for an ‘n’ stage LSFR the period of the sequence equals exactly $2^n-1$. Let us for example, take the initial state to be [1 0 0], where n =3. The clock pulse applied to the register causes the contents to be shifted to the right and the feedback is given to the left-most shift register. Figure 3.2 shows the output of the PN generator for an initial state [1 0 0]. The obtained output will be [0 0 1 0 1 1 1] with a period equal to $2^3 -1 = 7$, as shown in Figure 3.2. This output is periodic and the period of the sequence depends on the feedback connections.
Figure 3.1 PN Generator

<table>
<thead>
<tr>
<th>$R_1$</th>
<th>$R_2$</th>
<th>$R_3$</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>----</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 3.2 Output Table for PN Generator
3.1.1 Properties of PN Sequences

There are many properties of PN sequences that make it ideal to be used in watermarking schemes. PN sequences that have periods equal to \(2^n - 1\) for an \(n\)-stage LSFR satisfies the following properties:

- **Balance Property:** This property states that in the sequence generated the number of ones is equal to the number of zeros.

- **Run Property:** The word run is described as a sequence containing a single type of a digit. For a maximal length sequence one half of the runs are of length 1, one quarter of the runs are of length 2, and one eighth of the runs are of length 3. In general a sequence of length \(n\) will have exactly \(\frac{1}{2^n}\).

- **Shift Property:** This property states that for any ML sequence and its cyclically shifted sequence, the agreements and the disagreements among them will be approximately equal.

- **Cross-correlation Property:** Cross correlation is a standard method of estimating the degree to which two series are correlated. Consider \(x = \{x_0, x_1, x_2…x_{N-1}\}\) and \(y = \{y_0, y_1, y_2…y_{N-1}\}\) which denote two different pseudorandom sequences, the cross-correlation of these two sequences is defined as:

\[
R_k (x, y) = \frac{1}{N} \sum_{n=0}^{N-1} x_n y_{N-k}
\]

The two sequences \(x\) and \(y\) are said to be orthogonal if the cross-correlation between them is equal to zero and when \(x = \pm y\) then the cross-correlation is equal to \(N\).
• Autocorrelation Property: The autocorrelation properties of a pseudo random noise sequence are similar to the correlation properties of random noise i.e., it is single peaked. It is defined as:

\[ R_{ss}(k) = \frac{1}{N} \sum_{n=0}^{N-1} S_n \times S_n - k \]

where, \( R_{ss} \) is the autocorrelation of \( S \)

\( S_n \) is the cyclic shift by \( k \).

\[ R_{ss}(k) = -\frac{1}{2^{n}-1} \text{ for all } k \neq n \]

\[ R_{ss}(k) = 1 \text{ if } k = n \]

The MATLAB command \texttt{rand} is used to generate the pseudorandom numbers that are uniformly distributed. The sequence of numbers generated is determined by the state of the generator. This sequence will remain the same unless the state is changed. PN sequences have been used in watermarking due to its resistance to interference and its noise like characteristics. The watermark image, which is the information bearing data signal and the PN sequence as the spreading signal is applied to a product modulator. The resultant signal is a pseudorandom noise pattern that is added to the cover image to produce a watermarked image. Mathematically, the addition of a watermark is represented as:

\[ I_n(x, y) = k_1 \times w(x, y) + l(x, y) \]

\[ I_n(x, y) = a(x, y) \times b(x, y) + l(x, y) \]

where, \( a(x, y) = \text{Watermark Image} \)

\( b(x, y) = \text{d-sequence/Spreading signal} \)

\( l(x, y) = \text{Cover Image} \)
To recover the original watermark the watermarked image is multiplied again at the receiver with a pseudorandom noise sequence that is generated using the same seed that is used to generate the sequence. The unwanted noise signal is filtered out during the process of correlation by setting the threshold. The robustness of the watermarked image increases as the gain $k$ increases. But it should also be noted that there is a tradeoff between the values of $k$ and the quality of the image, which starts to degrade for higher values of $k$.

### 3.2 Decimal Sequences (d-sequences)

Decimal sequences are obtained when a number is represented in a decimal form in a base $r$ [18]. These sequences may terminate, repeat or be aperiodic. Any periodic pseudorandom sequence can be represented as a decimal sequence of a rational number. A certain class of d-sequences of the form $1/q$, $q$ being a prime number, exhibits the property where the digits spaced half a period apart add up to exactly $r-1$, where $r$ is the base in which the number has been expressed and for the same class of d-sequences, all the subsequences of length $m$, where $r^m > q$, are distinct. An upper bound for the nonzero values of the autocorrelation function of certain d-sequences has been established. The main properties of d-sequences are stated below and are discussed at length in [16], [19], [20]. Few schemes that employ d-sequences in watermarking have been implemented yielding good results [21], [22].

#### 3.2.1 Properties of d-sequences

The following theorems summarize the main structural properties of sequences representing the decimal expansion of a rational number:
• Theorem 1: We may express any positive number as a decimal number in the base \( r \) as,

\[ A_1A_2...A_{s\pm1}.a_1a_2... \]

where \( 0 \leq A_i < r, 0 \leq a_i < r \), not all \( A \) and \( a \) are zero, and an infinity of the \( a_i \) are less than \((r-1)\). There also exists a one to one correspondence between the numbers and the decimals and

\[ x = A_1r^s + A_2r^{s-1} + ... + A_{s+1} + \frac{a_1}{r} + \frac{a_2}{r^2} + ... \]

We can use the decimal sequences of rational and irrational numbers in order to generate pseudorandom sequences.

• Theorem 2: If \( q \) is a prime and \( r \) is a primitive root of \( q \), then the d-sequence for \( 1/q \) is termed as the maximum length decimal sequence (MLDS) in the base \( r \).

MLDS shall be represented by a string of their first \( q-1 \) digits, without showing the decimal, or as \((1/q)\). It is also known that for each prime \( q \) there exist \( \phi(q-1) \) maximum length sequences in different scales.

• Theorem 3: A MLDS \((1/q)\), when multiplied by \( p, p<q \), is a cyclic permutation of itself. Example: Consider \( x = (1/7) \). The decimal sequence of \( x \) in base 10 is maximum length because \( 10^2 \neq 1 \) (modulo 7), \( 10^3 \neq 1 \) (modulo 7) but \( 10^6 = 1 \) (modulo 7). The decimal sequence is 1 4 2 8 5 7, which corresponds to the remainder sequence 3 2 6 4 5 1.

The remainder sequence has considerable structure. Thus 3, 3^2, 3^3, 3^4, 3^5, 3^6 all computed modulo 7 yield the successive digits of the sequence. Now if \( x = (3/7) \) the remainder sequence starts with 30 = 2 (modulo 7) and in fact is 2 6 4 5 1.
3, and hence the decimal sequence for 3/7 is 4 2 8 5 7 1. This suggests that the structure of the remainder sequence must also show up in the decimal sequence.

- **Theorem 4:** If the decimal sequence in base $r$ of $p/q; (p, q) = 1, p < q$, and $(r, p) = 1$ is shifted to the left in a cyclic manner $l$ times, the resulting sequence corresponds to the number $p'/q, (p', q) = 1, p' < q$ where $p' = r' \times p \mod q$.

- **Theorem 5:** For a MLDS $(1/q) = a_1a_2...a_k, k = q-1$, in base $r$:

  \[ a_i + a \left( \frac{k}{2^{i+1}} \right) = r - l \]

  *Example:* Consider $x = (1/19)$ in base 2. Here the decimal sequence for $x$ is 0 0 0 0 1 1 0 1 0 1 1 0 0 1 0 1 and $a_i + a \left( \frac{k}{2^{i+1}} \right) = r - l = 1$.

- **Theorem 6:** The Hamming distance $d_j$ between the binary maximum length sequence $(1/q)$ and its $j^{th}$ cyclic shift satisfies

  \[ d_j \geq k / m, j \neq 0, j < k, \]

  where $2^m > q, k = q-1$. This shows that at least one of the each $m$ consecutive digits will be different. Therefore, the minimum distance between each set of $m$ digits is one. For a total of $k$ such group of digits, the distance will be $k$ and since we consider the sequence $m$ times over, the minimum distance is $k/m$. 

24
3.2.1.1 Cross–correlation Property

Let \( C_{12}(\tau) = \left( \frac{1}{N} \right) \sum_{i=1}^{N} a_i b_{i+\tau} \) represent the cross-correlation function of two maximum length sequences \( x = a_1 a_2 ... a_k \) and \( y = b_1 b_2 ... b_k \). The period of the product sequence \( a_i b_{i+\tau} \) is \( N = \text{LCM}(k_1, k_2) \) where LCM is the least common multiple.

- Theorem 7: The cross-correlation function of two maximal length sequences in the symmetric form is identically equal to zero if the ratio \( k_1/k_2 \) of their periods reduces to an irreducible fraction \( n_1/n_2 \) where either \( n_1 \) or \( n_2 \) is an even number.

3.2.1.2 Autocorrelation Property

The autocorrelation of the binary maximum length decimal sequence in the symmetric form satisfies \( C_1(j)\leq 1 - 2/\sqrt{m}, j \neq 0, j < k \). These sequences can be used for error detection and correlation and this is indicated by the existence of a lower bound between a sequence and its cyclic shifts. For a normal number, the autocorrelation function is defined as

\[
R_x(\tau) = E(a_n, a_{n+\tau})\ ,
\]

where the \( n^{th} \) digit of the sequence \( a_n \in \{0, 1, 2 ... r\} \). Since each of the digits occur with a frequency \( 1/r \) therefore \( R_x(0) = E(a_n^2) = (r-1)(2r-1)/6 \). Also for such a number the successive sequence of digits are independent and hence

\[
R_x(\tau) = E(a_n, a_{n+\tau}) = E(a_n)E(a_{n+\tau}) = (r-1)^2 / 4
\]
The autocorrelation function is two valued if the digits from zero to \((r-1)\) were mapped symmetrically about zero by the transformation \(a_i = 2a_i - (r-1)\), a straight forward calculation shows that

\[
R_x(\tau) = \begin{cases} \frac{(r^2 - 1)}{3} & \tau = 0; \\
0 & \text{otherwise} \end{cases}
\]

### 3.2.2 Generation of d-sequences

A simple algorithm is used to generate the binary decimal sequence. The hardware involved comprises of feedback shift registers that allow carry. This hardware is similar to the one used in the generation of m-sequences (pseudorandom shift register sequences). The equation to generate a binary sequence for \(1/q\) is shown below:

\[
a^i = (2^i \mod q) \mod 2
\]

The algorithm used for the generation is called the Tirtha algorithm and it is used whenever the prime number \(q\) is given in terms of radix \(r\) as \(q = tr-1\), where \(t\) is an integer.

- **Theorem 1:** Consider that \(1/ (tr-1)\) defines the decimal sequence \(a_1a_2a_3...a_k\), where \(r\) is the radix. Consider another sequence \(u_1u_2u_3...u_k\), where for all \(i, u_i < t\), then

\[
u_i + a_i = u_{i+1} + ta_{i+1}
\]

**Example:** Consider \(1/11 = 1/(2 \times 6 - 1)\) in base 2 when \(t = 6\).

<table>
<thead>
<tr>
<th>(u^{-1})</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(a^{-1})</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Hence the d-sequence corresponding to 1/11 is 0 0 0 1 0 1 1 1 0 1.

The hardware structure that is used for the generation of d-sequence for \( n \)-stages of shift registers is shown below:

![Decimal Sequence Generator Diagram]

Figure 3.3 Decimal Sequence Generator

The \( C \)'s represent the carries that are added to the immediate preceding stage. When the carry is generated by the extreme left stage, it is introduced into this stage at the very next clock instant. The inverse order of the sequence generated will be considered as the d-sequence. The number of stages of the shift register needed to generate the binary d-sequence is about \( \log_2 q \). The algorithm also works for non-binary sequences not of the form \( 1/(tr-I) \). In order to use the standard form we will need to multiply the given fraction by an appropriate integer \([18], [19]\). Certain primes are found to have low gains associated with them as the d-sequences generated by these primes have good statistical properties in terms of being random. Decimal sequences have some
advantages over pseudo-random sequences as they have zero autocorrelation for certain shifts and this helps in better watermark detection.

### 3.3 Proposed Scheme

Since the proposed watermarking scheme exploits aspects of human visual system, it may be called perceptual watermarking or content based watermarking. Our scheme utilizes the perceptual information of the image content, by taking advantage of frequency selectivity and assigns weights to provide some perceptual criteria in the watermarking process. This directly results in providing the watermark more invisibility. DCT based watermarking is resistant to compression and other frequency-based attacks, and this results in the scheme being very robust as well as imperceptible than most other schemes. We divide our scheme into three steps namely: 1) generation of a mask based on the perceptual properties of the image; 2) watermarking, by spreading the d-sequence in the frequency domain, by multiplying it with the weights calculated from step1; and 3) extraction of the watermark by using a correlation-based method.

#### 3.3.1 Just Noticeable Distortion (JND) Visual Mask

Just Noticeable Distortion is defined as a measure referring to the capability of a human observer to detect noise or distortion in the field of view. The image is first analyzed both in the frequency domain as well as spatial domain to detect the distortion sensitivity of the image based on its content. Most schemes regard the process of watermarking as adding noise to an image. An image can be distorted only to a certain limit without making the difference between the original image and the watermarked one perceptible [1]. The limit to which we can alter a pixel value without making it
perceptible is the JND. There are many characteristics that define the JND, of which we consider, texture, luminance, edge and corners to estimate a mask, which is the weight assigned to the particular block. This weight is used to modulate the watermark thereby keeping the amplitude of the signal below the noise distortion sensitivity of each pixel.

Our model takes into account an image independent part based on frequency sensitivity and an image dependent part based on edge and corner information, and the luminance sensitivity [23] in designing the perceptual mask. We first segment the image into blocks based on the frequency characteristics, as the human eye is sensitive to certain frequencies more than the others. In other words when we perform the DCT on the image the resultant will be DCT coefficients arranged in a specific order based on the frequency. The image characteristics that are considered to generate the mask are texture, edge, corner and luminance. Several studies on the HVS have shown that in highly textured areas the distortion visibility is low. These areas are suited to hide the watermark signal and therefore the JND values corresponding to those areas must be high. Edge, corner and the luminance sensitivity values that are generated from the spatial domain are considered as equally important characteristics that influence the human perception of images and they have the lowest JND values [1].

Watermark strength depends on the DCT coefficients of the original image. This provides adaptability, by allowing more watermark information to be embedded in blocks that have high texture. Image blocks having many edges or corners are assigned lower JND values because in these blocks the watermark can be more easily perceived. In theory, the definition of a good JND mask would depend on the accurate extraction of the luminance, texture, edge and corner information from the image as this will provide
maximum strength/robustness, high capacity and imperceptibility. Our scheme is image adaptive as it incorporates the local information extracted from the image. The algorithm that is used to generate the JND mask is explained below:

Let \( f(x, y) \) be the original grey scale cover image. This image is segmented into non-overlapping blocks of size \( 8 \times 8 \). This is denoted as \( B_k, n = 0,1,2,...,N - 1 \).

\[
f(x, y) = \bigcup_{n=0}^{N-1} B_n = \bigcup_{n=0}^{N-1} f_n(i, j), \text{ where } 0 \leq i, j < 8
\]

The MATLAB command \( dct2 \) is used to perform the operation of converting each \( 8 \times 8 \) block into its respective block consisting of DCT coefficients, which concentrates information based on frequency, making it especially useful for compression applications. The formula to calculate a two dimensional DCT in MATLAB is given by:

\[
B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos \left( \frac{(2m+1)p}{2M} \right) \cos \left( \frac{(2n+1)q}{2N} \right), \quad 0 \leq p \leq M - 1, \quad 0 \leq q \leq N - 1
\]

\[
\alpha_p = \begin{cases} 
\frac{1}{\sqrt{M}} & p = 0 \\
\frac{2}{\sqrt{M}} & 1 \leq p \leq M - 1
\end{cases}
\]

\[
\alpha_q = \begin{cases} 
\frac{1}{\sqrt{N}} & q = 0 \\
\frac{2}{\sqrt{N}} & 1 \leq q \leq M - 1
\end{cases}
\]

where \( M \) and \( N \) are the row and the column size of \( A \) respectively.

Each element of the two-dimensional array of frequency components represents a two-dimensional frequency component. The element in the upper-left corner is the DC coefficient for the entire array and all the remaining coefficients containing the frequency information are called the AC components. This arrangement is shown in Figure 3.4. The DC coefficient is proportional to the average pixel value in the original block and the AC
coefficients in a block describe their variation around the DC value. The coefficients close to the DC component represent the highly correlated pixel values i.e., the low frequency, while the coefficients towards the lower right corner represent the high frequency such as the edges and the noise.

![Figure 3.4 Definition of DC/AC components for a 8×8 DCT block](image)

**Table 3.1** DCT values for a 8×8 block

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-63.809</td>
<td>2.21896</td>
<td>-0.22811</td>
<td>16.4982</td>
<td>6.036879</td>
<td>14.77166</td>
<td>0.671012</td>
<td>-3.07591</td>
<td></td>
</tr>
<tr>
<td>49.55688</td>
<td>34.24766</td>
<td>12.33623</td>
<td>3.363575</td>
<td>-3.44614</td>
<td>-0.86883</td>
<td>0.463388</td>
<td>-11.3807</td>
<td></td>
</tr>
<tr>
<td>33.50481</td>
<td>-7.30546</td>
<td>2.138439</td>
<td>-5.88645</td>
<td>5.835287</td>
<td>-1.48819</td>
<td>2.109087</td>
<td>-0.12882</td>
<td></td>
</tr>
<tr>
<td>-2.60643</td>
<td>-10.2655</td>
<td>-3.5449</td>
<td>2.728022</td>
<td>0.875</td>
<td>1.889503</td>
<td>-0.69772</td>
<td>1.97989</td>
<td></td>
</tr>
<tr>
<td>3.552485</td>
<td>5.403749</td>
<td>2.498121</td>
<td>4.388331</td>
<td>0.419913</td>
<td>2.949265</td>
<td>5.629466</td>
<td>6.403936</td>
<td></td>
</tr>
<tr>
<td>-4.15595</td>
<td>-5.16516</td>
<td>-0.28661</td>
<td>3.61801</td>
<td>-2.76683</td>
<td>0.959746</td>
<td>-2.33623</td>
<td>6.484752</td>
<td></td>
</tr>
<tr>
<td>0.026231</td>
<td>1.300611</td>
<td>0.916132</td>
<td>0.8835</td>
<td>3.914792</td>
<td>-0.38353</td>
<td>2.654941</td>
<td>7.218225</td>
<td></td>
</tr>
</tbody>
</table>
Texture: It is defined as the visual quality of the surface of the object, exposed in an image by variances in tone, depth and shape. On obtaining the DCT coefficients, we first extract the texture information directly by analyzing these coefficients. This information is calculated from the DCT coefficients i.e., derived from the visual model consisting of an image independent part based on frequency sensitivity. Each $8 \times 8$ block which consists of the 64 DCT coefficients is analyzed and as we know that the highly textured regions or along edges the energy of the signal is concentrated in the high frequency components while in areas where the image is uniform the energy of the signal is concentrated in the low frequency components. To determine a measure for the texture information within each block based on the energy in the AC coefficients we use the formula:

$$P_T = \log(\sum_{i=1}^{63} v_i^2 - v_0^2)$$

where $v_i, i = 0, 1...63$ are the 64 DCT coefficients of the $8 \times 8$ block that is being considered. We must note that $v_0$ is the value of the DC component of the DCT coefficients and it is not considered when calculating the texture value. For each block the obtained values of $T_E$ are first scaled to the range of $[0, 64]$ and then the normalized values are assigned to the corresponding blocks.

$$M_T = \frac{64 \times P_T}{\max(P_T)}$$

Hence for an image matrix of size $512 \times 512$ we will have a matrix $T_E$ of size $64 \times 64$ where each one of those values corresponds to the texture information of each $8 \times 8$ block.
Edge: Edges are extracted from the pixel domain and this information is useful in determining the amount of watermark information that can be embedded in the image. We need an algorithm that accurately extracts the edge information from the image and by accurate we mean something that not only shows, as many edges as there are present but also differentiate between real edges and spurious edges, which may occur due to noise and texture. Human eye is more sensitive to changes in areas having more edges as compared to those with lesser or no edges (smooth areas) and hence we will use this information to assign a weight accordingly. In order to perform this we will make use of an algorithm implemented by Peter Kovesi \cite{24}, \cite{25} that has been proved to extract edges better than most other edge detection algorithms.

There are many methods of finding the edges based on either the gradient of the image $I (u, v)$ or the zero crossings, after filtering the image $I (u, v)$ with a Gaussian or a Laplacian filter. One of the better algorithms returning many more edges with good accuracy in determining spurious edges is based on the phase congruency of feature detection \cite{24}. This method is invariant to image contrast, unlike most methods. Phase congruency is described as a dimensionless quantity that provides the information that does not change with image contrast and this is used to determine the principal magnitudes of moments of phase congruency. An edge would be one where the maximum moment of phase congruency is large. More information on phase congruency and extraction of edges using phase congruency can be found in \cite{25}. This algorithm can also be used to detect a corner map, which is strictly a subset of the edge map. Although corners are a part of an edge it is found that most edge detectors do not accurately detect edges at a corner and hence we will use a separate algorithm to detect the corners. We are
concerned with determining as many edges and corners without one being dependent on the other and the rationale behind this, is explained later in this chapter. We will take an average of these two values to estimate the JND mask. Using a binary edge map, we calculate the normalized edge information for each block using the formula:

\[ M_E = \frac{64 \times P_E}{\max(P_E)} \]

where \( P_E \) is the cardinality of set of pixels at edge locations in each block while \( \max(P_E) \) is the maximum value of \( P_E \) over the entire image. Detected edges of the image Lena, by the two different methods mentioned earlier are shown in Figures 3.5 and 3.6. We clearly observe from these two images that the number of edges that are detected using phase congruency is far more than those detected by the Canny operator. It is also more accurate in distinguishing between real and spurious edges.

Figure 3.5 Edge Extraction using Canny Operator
Corner: Another important aspect in the pixel or spatial domain is information pertaining to the corners. Corners have long been recognized as visual information carriers and various algorithms have been proposed to detect them for use in basic visual tasks [26]. Corners are considered more localized than edges and are better in defining shapes of objects in images as an edge can provide local information only in one direction, normal to the edge [26]. A corner represents the point where two edges meet and the human eye is more sensitive to changes made in these places. Our aim is to determine a factor that employs both the edge and the corner information accurately in order to determine the final JND mask. Perceptual watermark schemes consider
uniformity as an important factor in human perception. Kay and Izquierdo make use of a Moravec operator to extract uniform regions. It is essentially a corner detector that uses a sliding window approach to detect the smoothness in a block with the help of intensity variation. The number of pixels belonging to a uniform area in a block is regarded as the uniformity factor [1]. But a Moravec operator is found to identify false corners especially at isolated pixels, due to its sensitivity to noise [27]. The number of corners in a block accurately represents the uniformity factor and we will utilize an improved corner detector to determine this factor. There are many algorithms using different approaches to detect the right corners while eliminating the false corners and we need an algorithm that detects all the true corners that are present in an image accurately, reducing the probability of detecting false corners. The algorithm should be robust with respect to noise and the corner points should be well localized. We make use of an improved corner detection algorithm based on curvature scale space (CSS) with adaptive threshold and dynamic region of support in order to detect corners in the image [28], [29].

Methods employing CSS to detect corners have been very successful and it is believed to perform better than the existing corner detectors [27], [28]. The main steps involved in corner detection are listed below [28]:

- Extracting the edge information/contours from a binary edge map that is obtained using any good edge detection method, in our case, is by using the algorithm by Peter Kovesi.
- Filling in the gaps in the contours.
- Computing the curvature at a fixed low scale to retain all the true corners.
Finally, the curvature local maxima are considered as corners while eliminating the rounded and false corners resulting from noise using adaptive threshold and the angle of corner.

Figure 3.7 shows the effective detection of corners for a Lena image based on the curvature scale space.

![Corner Detection using Curvature Scale Space Method](image)

Figure 3.7 Corner Detection using Curvature Scale Space Method

On obtaining the corners by the above method we calculate the corner information for each block of the image using the formula:

\[
M_c = \frac{64 \times P_c}{\max(P_c)}
\]
where \( P_c \) is the cardinality of the group of pixels determined to be corners in each block, \( \max ( P_c ) \) is the maximum value of \( P_c \) over all the blocks in the image and \( M_c \) is the normalized value of the corner information.

Luminance: It is defined as the way the human eye perceives brightness of different colors. This property influences the perception of the image information by the human eye. This factor may be determined in two different domains, the frequency domain where the DC component of the DCT coefficients is used as developed by the Watson model [30], and the pixel domain. The DC component carries significant information with respect to the luminance in the block and it determines the average brightness in the block. This mean value of luminance of a local block is estimated by the formula:

\[
D_L = \left( \frac{DC_{b}}{DC_{mean}} \right)^{\alpha}
\]

where \( DC_b \) is the DC coefficient of the DCT for block \( b \)

\( DC_{mean} \) is the DC coefficient of the mean luminance of the display

\( \alpha \) is the parameter that is used to control the luminance sensitivity

The value of \( \alpha \) is set to 0.649 as per the model used by the authors [31]. The luminance factor is generated in both the frequency domain as per the Watson model as well as in the pixel domain and hence it is partly image dependent i.e., it is obtained by analyzing the pixels in the block. Our scheme utilizes the luminance factor that is calculated by measuring the average pixel value of the gray scale image for that block.

\[
M_L = \frac{P_L}{64}
\]
where $P_L$ is the sum of all the pixel values in the block and $M_L$ is the average of the luminance values within the considered block. The factors obtained from the edges and the corners and the luminance values of the image put together are called as the spatial masking values, as they are obtained directly by analyzing the pixels in the spatial domain. After obtaining the four values corresponding to the texture, edge, corners and the luminance we generate the initial mask using the equation:

$$J_1 = M_T - \frac{1}{2}(M_E + M_C)$$

The human vision system is more sensitive to the changes in intensity in the mid-gray region and it is to be noted that this sensitivity fails parabolically at both ends of the gray scale. Hence a correction to the initial JND parameter value is introduced and the final JND parameter value for each block is calculated as:

$$J_f = J_1 + (128 - M_L)^2$$

where $J_f$ is the initial JND parameter value, $M_L$ is the average of the luminance values within the considered block. An alternate method is to multiply the luminance factor $D_L$, derived in the frequency domain, with the JND value $J_1$ generated above and the DCT coefficient, at the time of watermark embedding. Effectively the JND value for each block is obtained by analyzing properties like texture from the frequency domain and some basic properties derived from the spatial domain namely, edge, corner and the average luminance value.

### 3.3.2 Watermark Embedding

We then perform the watermark insertion in DCT domain by modifying selected DCT coefficients, which embeds a d-sequence based on the watermark, for each block.
The JND value controls the strength of watermark for each block. In other words, the strength of the watermark component embedded, in a block with a low JND value, is less. This is because any changes made in this block are more perceptible to the human eye. On the other hand, the strength of the watermark component embedded in a block with a high JND value is low.

The whole idea of block based JND watermarking is to incorporate the local perceptual masking effects as it provides local control of the strength of watermark based on the image content. Experimental results have shown that embedding the watermark in the high frequency components that carry less perceptually significant information results in the removal of the watermark through compression and noise attacks, while adding it in the low frequency components, which carries perceptually important information, results in visible changes in the watermarked image [14]. In our scheme we will select and modify only those DCT coefficients that lie in the mid-frequency band.

The location of mid-frequency components of a $8 \times 8$ DCT block is shown below in Figure 3.8. The first value is called the DC component of the image and its DCT coefficient is relatively very high as compared to all the other coefficients, which are called the AC coefficients. Also the AC coefficients closer to the DC value comprise of the low frequency components while the ones at the bottom right are the higher frequency components.

The DC component of a DCT block is considered to carry perceptually significant information. It is believed that the DCT coefficients in the mid-frequency band have similar magnitudes. For each $8 \times 8$ transformed block the $d$-sequence multiplied by a scaling factor and the JND mask is added into the selected mid-frequency
DCT components while the low and high frequency coefficients are copied over unaffected.

Figure 3.8 Definition of DCT regions

Our scheme also differs from that of Kay and Izquierdo’s scheme in the way the random sequence is added to the image. The d-sequence is dependent on the actual watermark i.e., each DCT block of size $8 \times 8$ corresponds to either a 1 or 0 of the watermark bit and the d-sequence is added to the block if the watermark bit is 0 and it is subtracted wherever the watermark bit is 1. This increases both the robustness and the capacity of image to carry more watermark information while keeping the PSNR constant as compared to other spatial CDMA spread-spectrum watermarking methods where the distortion of the watermarked image increases exponentially with the size of the watermark.

The scaling factor denotes the strength of the watermark and it can be used to control the overall robustness and the quality of the image. Increasing this value increases
the strength of the watermark but introduces a gradual visible distortion while decreasing this value would result in better hiding of the watermark and hence better quality but decreases the strength of the watermark. An optimal value needs to be decided upon depending on the watermark and the d-sequence before embedding. Upon inverse transformation the watermark will be scattered over the entire image and we obtain the watermarked image. The watermark embedding is done using the formula:

\[
I_w(u,v,b) = \begin{cases} 
I(u,v,b) + (\beta \times J_F(b) \times d) & u,v \in F_{mid} \\
I(u,v,b) & u,v \in F_{mid}
\end{cases}
\]

\[
Watermark \ bit = 0
\]

\[
I_w(u,v,b) = \begin{cases} 
I(u,v,b) - (\beta \times J_F(b) \times d) & u,v \in F_{mid} \\
I(u,v,b) & u,v \in F_{mid}
\end{cases}
\]

\[
Watermark \ bit = 1
\]

where \( I_w(u, v, b) \) is the modified DCT coefficient in location \((u, v)\) for block \(b\)

\( I(u, v, b) \) is the DCT coefficient in location \((u, v)\) for block \(b\)

\( \beta \) is the scaling factor

\( J_F(b) \) is the JND value generated for the block from the equation above

\( d \) is the d-sequence generated

\( F_{mid} \) is the middle frequencies of the DCT block

Finally, the block containing the watermarked DCT coefficients is inverse-transformed to obtain the final watermarked image. Each block containing the watermarked coefficients in the transformed domain is converted back to the image block.
in the pixel domain. Hence we obtain the final watermarked image. The algorithm used in MATLAB to compute the inverse DCT is shown below:

\[
A_{mn} = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \alpha_p \alpha_q B_{pq} \cos \frac{\pi (2m+1)p}{2M} \cos \frac{\pi (2n+1)q}{2N}, \quad 0 \leq m \leq M - 1, \quad 0 \leq n \leq N - 1
\]

\[
\alpha_p = \begin{cases}
\frac{1}{\sqrt{M}} & p = 0 \\
\frac{2}{\sqrt{M}} & 1 \leq p \leq M - 1
\end{cases}
\]

\[
\alpha_q = \begin{cases}
\frac{1}{\sqrt{N}} & q = 0 \\
\frac{2}{\sqrt{N}} & 1 \leq q \leq N - 1
\end{cases}
\]

3.3.3 Watermark Detection

In order to recover the watermark we use the correlation-based watermark detection scheme. Here, the image is first broken down into the same 8×8 blocks as done in watermark embedding and then the DCT is performed on each block. The DCT coefficients of the mid-frequency values thus obtained are compared with the d-sequence that is generated using the same prime number used in watermark embedding.

Correlation \( C(b) = \frac{1}{N} (I^*(b) W(b)) \)

Recovered watermark bit = \[
\begin{cases}
0 & \text{if } C(b) > T \\
1 & \text{if } C(b) < T
\end{cases}
\]

where, \( T \) is the Threshold level

\( C(b) \) is the correlation value for block \( b \)
$I^* (b)$ is the DCT coefficient of the watermarked image assumed to have been transformed by processing or attack.

$W (b)$ is the d-sequence that is generated using the same prime number that was used to generate the d-sequence at the embedding stage.

Selecting a threshold $T$ in the process of correlation filters out the unwanted noise. We must be aware that determining the presence of a watermark through correlation is a statistical test and hence there is a possibility of obtaining detection errors. Errors can be of two types, ‘0’ that is falsely detected as ‘1’ and ‘1’ that is falsely detected as ‘0’. The setting of the threshold $T$ is considered as a decision based on the need to minimize errors, such as those mentioned above, in watermark detection. The results of our watermarking scheme are shown in the next chapter. Some watermarking attacks are conducted to test the performance of the proposed watermarking scheme.

### 3.3.4 Watermark Evaluation

Signal to noise ratio (SNR) effectively measures the quality of the watermarked image as compared to the original image. This difference is represented as an error function that shows how close the watermarked image is to the original image and it is written as:

$$e(x, y) = I(x, y) - I_w(x, y)$$

The larger the value of $e (x, y)$ the greater is the distortion caused by the watermark and the attacks. One of the simplest distortion measures is the mean square error (MSE) function [32]:

$$MSE = \frac{\sum [f(i, j) - F(i, j)]^2}{N^2}$$
The peak signal to noise ratio (PSNR) is calculated by using the formula:

$$PSNR = 20 \log_{10} \left( \frac{255}{\sqrt{MSE}} \right)$$

$\sqrt{MSE}$ is called the root mean square error and 255 is the maximum value of luminance level. It should be noted that PSNR does not take aspects of the HVS into consideration although it provides an overall evaluation of the difference between the original and the watermarked image. For this reason, we will use another perceptual quality measure called the weighted peak signal to noise ratio (WPSNR). This metric takes into account the objective measure as well as the HVS. The human eye is less sensitive to changes in highly textured areas and hence an additional parameter called the noise visibility function (NVF) is introduced. This helps us calculate the change in the perceptual quality more accurately. The formula for WPSNR is shown below:

$$WPSNR = 20 \log_{10} \left( \frac{255}{\sqrt{MSE \times NVF}} \right)$$

NVF uses a Gaussian model to estimate the amount of texture content in any part of an image [32]. In regions with edges and texture NVF will have a value greater than 0 while in smooth regions the value of NVF will be greater than 1. The formula to calculate this factor as a simplified function is:

$$NVF = NORM \left( \frac{1}{1 + \delta_{block}^2} \right)$$

where $\delta$ is the luminance variance for the $8 \times 8$ block and NORM is a normalization function.
Chapter 4

Experimental Results

Our experiments on the proposed content based watermarking are based on grayscale images. The cover objects used are the images of Lena and Boat. The prime number $q$ that is used to generate the d-sequence is 2467 and on using the appropriate scaling factor and threshold we notice that the watermark is recovered perfectly well. As mentioned earlier the scaling factor is set according to the content and the quality of the watermarked image. The greater the scaling factor, the better is the watermark detection however reducing the overall quality of the image. Hence an optimum value is chosen accordingly and it is set to 0.007 for Lena. Here, a watermark of size $12 \times 12$ pixels has been used and the peak signal to noise ratio (WPSNR) is found to be 38.99 dB.

Figure 4.1 Lena Reference Image (512 x 512 Pixels)
Figure 4.2 Watermarked Image WPSNR = 38.99 dB

Figure 4.3 Original Watermark

Figure 4.4 Recovered Watermark

Figure 4.5 Sufficient Statistic for $q = 2467$
On changing the prime number $q$ to 8069 and using the same scaling factor as above we notice that the original watermark has been recovered perfectly. When we use a slightly bigger watermark of size $15 \times 12$ pixels and peak signal to noise value (WPSNR) is again found to be 38.99 dB, as shown in Figure 4.6. Note that the scaling factor is kept the same, as increasing it would result in some visible distortions in the watermarked image. The WPSNR will be the same for a given scaling factor; a change in the watermark size will not affect this value because all the DCT blocks are modified irrespective of the size of watermark, providing robustness and easy watermark detection.

The watermark is added in such a way so as to keep the amplitude of the signal below the noise distortion sensitivity of each pixel and varies very slightly with the size of the watermark for a given image. The detection statistic does not change much for the two images because there is minimal change in the size of the watermark.

Figure 4.6 Watermarked Image WPSNR =38.79 dB
The correlation plot shown above indicates that the margin of error is very less for the detection statistic. Now, we use a different cover image with the same two prime numbers that was used for the earlier result; the cover image is Boat (Figure 4.10), and watermarks are of sizes $32 \times 32$ and $64 \times 64$. We observe that there is a change in the value of WPSNR for Boat as this image has a greater scaling factor. This value is decided by observing the quality of the watermarked image. The scaling factor for every image is...
dependent on the image content and varies according to the properties that are used to determine the mask namely, texture, corner, edge and luminance information.

For this image the scaling factor suggested is 0.084. The watermark is recovered, although some amount of noise is present in the recovered watermark and it is seen that this scheme holds good for different sizes of watermark. As observed, the scaling factor that is used to watermark the cover object allows enough information to be added without any visible distortion in the final watermarked image while also allowing for a good recovery of the watermark.

Figure 4.10 Watermarked Image WPSNR = 35.54

Figure 4.11 Embedded Watermark

Figure 4.12 Recovered Watermark
In the figure shown below the same cover object is embedded with a watermark of size $64 \times 64$ and as we observe, the watermark is recovered, again with some noise due to detection.

Figure 4.13 Watermarked Image WPSNR = 35.54

We experimented with various combinations of primes to verify the robustness of our scheme and found that in almost all the combinations the detection of the watermarks was highly satisfactory. The WPSNR value for the relatively smaller watermarks has
been found to be same and it slightly decreases for larger watermarks. Similarly the WPSNR value remains constant for all the prime numbers. The correlation threshold is entirely dependent on the scaling factor, which in turn is dependent on the image content. This factor is fixed for an image and value is fixed based on the desire to minimize false alarms and false rejections. The results for different watermark sizes and primes are shown in Table 4.1.

Table 4.1 WPSNR table for different primes

<table>
<thead>
<tr>
<th>Image</th>
<th>Watermark</th>
<th>Scaling Factor</th>
<th>WPSNR (dB)</th>
<th>Prime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>1</td>
<td>0.07</td>
<td>38.99</td>
<td>2467</td>
</tr>
<tr>
<td>Lena</td>
<td>2</td>
<td>0.07</td>
<td>38.99</td>
<td>5647</td>
</tr>
<tr>
<td>Lena</td>
<td>3</td>
<td>0.07</td>
<td>38.99</td>
<td>10459</td>
</tr>
<tr>
<td>Lena</td>
<td>4</td>
<td>0.07</td>
<td>38.99</td>
<td>2999</td>
</tr>
<tr>
<td>Boat</td>
<td>1</td>
<td>0.082</td>
<td>35.54</td>
<td>5647</td>
</tr>
<tr>
<td>Boat</td>
<td>2</td>
<td>0.082</td>
<td>35.54</td>
<td>2467</td>
</tr>
<tr>
<td>Boat</td>
<td>3</td>
<td>0.082</td>
<td>35.54</td>
<td>1109</td>
</tr>
<tr>
<td>Boat</td>
<td>4</td>
<td>0.082</td>
<td>35.54</td>
<td>8069</td>
</tr>
</tbody>
</table>

Figure 4.16 Watermark (12×12)  

Figure 4.17 Watermark (15×12)  

52
The change in WPSNR values with varying scaling factors for images Lena and Boat may be observed in Figure 4.20. From the plot we see that the rate at which the PSNR decreases for increasing values of the scaling factor is unchanged but the huge gap between the values suggest the difference in the JND values due to the image content. The WPSNR for the scaling factor used for Boat is 35.54, but in the case of watermarking Lena, for the same value the image is highly distorted.

Figure 4.18 Watermark (32 × 32)  
Figure 4.19 Watermark (64 × 64)

Figure 4.20 WPSNR vs. Scaling Factor Plot for Lena and Boat
Table 4.2 WPSNR table for different scaling factors

<table>
<thead>
<tr>
<th>Scaling Factor</th>
<th>WPSNR (dB)</th>
<th>Lena</th>
<th>Boat</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.007</td>
<td>38.9938</td>
<td>56.9153</td>
<td></td>
</tr>
<tr>
<td>0.009</td>
<td>36.8109</td>
<td>54.7324</td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>35.8958</td>
<td>53.8173</td>
<td></td>
</tr>
<tr>
<td>0.02</td>
<td>29.8752</td>
<td>47.7967</td>
<td></td>
</tr>
<tr>
<td>0.03</td>
<td>26.3533</td>
<td>44.2748</td>
<td></td>
</tr>
<tr>
<td>0.04</td>
<td>23.8546</td>
<td>41.7761</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td>21.9164</td>
<td>39.8379</td>
<td></td>
</tr>
<tr>
<td>0.06</td>
<td>20.3327</td>
<td>38.2542</td>
<td></td>
</tr>
<tr>
<td>0.07</td>
<td>18.9938</td>
<td>36.9153</td>
<td></td>
</tr>
<tr>
<td>0.08</td>
<td>17.834</td>
<td>35.54</td>
<td></td>
</tr>
<tr>
<td>0.09</td>
<td>16.8109</td>
<td>34.7324</td>
<td></td>
</tr>
<tr>
<td>0.10</td>
<td>15.8958</td>
<td>33.8173</td>
<td></td>
</tr>
</tbody>
</table>

Two different cover objects have been used to observe the differences in various properties that are used to calculate the JND mask. The normalized texture plot for Lena, for two randomly selected rows, each consisting of 64 blocks is shown in Figure 4.21. We observe that a block with more texture has a higher value than one with less texture and hence more watermark information can be embedded at these locations without being perceptually evident.
A similar normalized plot for the total number of edges and corners present in two different rows in Lena is shown in Figure 4.22 and Figure 4.23 respectively. It is observed that the presence of both edges and corners in row 32, which is roughly at the center of the image, is slightly more than in first row located at the top of the image. The luminance plot for Lena is shown in Figure 4.24. It is seen that this plot has less number of peaks in the first row as compared to row 32 proving that there is more variation of luminance resulting in more peaks at roughly the center of the image, while the image of uniform luminance at the top.
Figure 4.22 Normalized Edge plot for Lena

Figure 4.23 Normalized Corner Plot for Lena
Figure 4.24 Normalized Luminance Plot for Lena

Figure 4.25 Normalized JND values for Lena
The JND mask value for two randomly selected rows for Lena is shown in Figure 4.25. This plot signifies the maximum weight of the watermark allowed to be added to the image without a visible distortion. A similar plot for the same two rows for the Boat image is shown in Figure 4.26. We can notice the change in this plot for these two images. It is also seen that the values are equally high for both Lena and Boat in row 32 and hence more watermark information can be embedded in these areas.

Figure 4.26 Normalized JND values for Boat

Most frequency-based techniques are considered to be robust against compression and filtering. Since, the watermark is spread throughout the cover object based on the frequency; an attack that is aimed at the watermark removal cannot be successful unless it
attacks the fundamental structure of the image itself, which would result in a highly degraded image. In DCT watermarking, the watermark is spread over the entire spectrum therefore making it highly robust to frequency-based attacks. Since the watermarking is performed only to the mid-frequency components of the image blocks, the effect of most compression algorithms that usually target the high frequency components is largely avoided and hence the high robustness against compression.

4.1 Attacks and Analysis of Results

Various attacks are performed to test the robustness of our scheme and it is found, that our scheme performs excellently against JPEG compression and moderately well against common spatial attacks. The block size was kept constant at 8×8, mostly in anticipation of the JPEG compression. Embedding the watermark in the same domain where the attack might possibly take place would result in a robust scheme and this can be taken advantage of. Similarly the payload as well as the robustness can be increased by also embedding in either the low or the high frequency components of the DCT block, depending on the attack.

We have made selective use of Stirmark benchmarking technique [33], [34], to test the robustness of our scheme for JPEG compression and median filter attack. The other attacks that are performed on the watermarked images are introduction of Gaussian and salt and pepper noise and use of image filter. The result of the JPEG compression for different quality factors \((q)\) ranging from 45 to 30 is shown below along with the recovered watermark. It should be noted that this scheme performs excellently for JPEG compression of quality factor 45 and above.
Figure 4.27 JPEG Compression ($q = 45$)

Figure 4.28 Recovered Watermark

Figure 4.29 JPEG Compression ($q = 40$)

Figure 4.30 Recovered Watermark
One of the most interesting results is that the recovered watermark after JPEG compression with a quality factor of 45 is sometimes much better than the watermark that is recovered from a pristine watermarked image. The reasoning behind this is that, the detection errors in the unmodified source are deemed to be right on the correlation boundary and the addition of noise is just enough to push them over the edge [14].

From the above results we can say that for JPEG compression with a quality factor of 40, the watermark detection and extraction is near perfect. The recovered watermark for a quality factor of 35 shows a number of detection errors and this only becomes highly noticeable for a quality factor of 30. The overall robustness of our scheme for JPEG compression is considered high level, according to the robustness requirements table provided by Petitcolas [34].

We then test our scheme for its robustness against different types of noise. This is done by first introducing noise into the watermarked Lena of size 512 × 512. Gaussian noise with zero mean is introduced to verify as to what extent our proposed scheme can withstand noise. From the results shown below, we can observe that for a Gaussian noise of 2 %, the watermark recovery is moderate, with very few detection errors. We must keep in mind that most DCT block based schemes offer moderate robustness to noise and less robustness to common geometric attacks like scaling. An alternate technique would be to employ dual watermarking, in both frequency as well as spatial domain using CDMA spread spectrum. But this would not serve the purpose of imperceptibility of the watermark and as it is widely known; CDMA spread spectrum techniques result in visible distortion, have limited capacity and high processing requirements [14]. This is due to the
exhaustive search performed to detect the embedded sequence over each pixel of the image.

Figure 4.35 Uniform Gaussian Noise 2%  Figure 4.36 Recovered Watermark

Figure 4.37 Salt 'n' Pepper Noise, Variance 0.01  Figure 4.38 Recovered Watermark
When we introduce salt and pepper noise with zero mean and a variance of 0.01 in the watermarked image, the watermark is recovered with a few detection errors as shown in Figure 4.37.

Another common attack on images is filtering. We test our watermarking scheme for its robustness to median filtering attack generated using Stirmark [33], [34]. Median filter is similar to an averaging filter; each pixel output is set to the median of the pixel values in the neighborhood of the corresponding input pixel, as specified by the window size. The window size of $3 \times 3$ is used for our experiments. This is considered as moderate robustness for any watermarking scheme. As we can see, the watermark has been recovered almost perfectly except for some detection errors, which are introduced due to the filter.

Figure 4.39 Median Filter ($3 \times 3$)  
Figure 4.40 Recovered Watermark
Contrast enhancement filter is commonly used as an image-processing tool. The MATLAB command `fspecial` is used to create an unsharp filter from the negative of the Laplacian filter with a parameter alpha. The value of alpha must be in the range 0 to 1 and this parameter controls the shape of the Laplacian. We tested our scheme for its robustness to contrast enhancement filtering and the watermark recovery is found to be very good as shown above.
Chapter 5

Conclusions

This thesis provides a comprehensive evaluation and implementation of a content-based watermarking scheme that improves upon the earlier work of Kay and Izquierdo. By analyzing the cover object in both frequency and spatial domains, a distortion sensitivity of the image content is determined. Local information that is derived from properties such as texture, corner, edge and luminance is used to determine a mask of just noticeable difference values. This value, which is based on the image content, determines the strength of the watermark information that will be embedded. Our observations regarding the proposed watermarking scheme are summarized below:

- We by employing a better method of detecting edges using phase congruency, allowing us to detect more edges accurately.

- Our scheme implements a better corner detection algorithm that detects corners using curvature scale space instead of a Moravec operator. The detected corner is used as a factor to establish the uniform regions in the image, which is utilized to determine the JND mask.

- The robustness of our scheme to JPEG compression is found to be very good at a quality factor of 40 and reasonably good at a quality factor of 35. The results for other image processing attacks like median filtering and contrast-sharpening filter were also found to be good, although it is not very robust against scaling and high noise levels.
• Our scheme introduces a content based watermarking scheme using decimal sequences and the results are found to be highly satisfactory in terms of watermark detection. Any random sequence may be used to embed the watermark and the decision of using decimal sequences is based on the ease with which it can be generated, requiring only a prime number. Also, more flexibility can be achieved with the choice of various prime numbers that can be used for this purpose.

• A very good balance between robustness and imperceptibility has been achieved using this scheme as observers can evaluate the quality of the watermarked image as well as the recovered watermark to be good. Experimentation using various sizes of watermarks and different images enables a better understanding of the scheme. WPSNR is used to evaluate the perceptual quality of the watermarked image effectively and accurately, considering the effect of HVS.

Although this thesis was limited to watermarking of gray scale images in the DCT domain, further research can be done in implementing content based watermarking using decimal sequence for color images and video watermarking. To increase the scheme’s robustness against geometrical attacks like scaling, cropping as well as higher noise levels, we suggest an implementation of a hybrid watermarking scheme in both the spatial as well as the frequency domain, which need not be restricted to DCT. CDMA spread spectrum approach to the hybrid scheme may also be considered although this may not quite serve the purpose of imperceptibility of the watermark.
Bibliography


Vita

Arvind K. Parthasarathy was born in Secunderabad, Andhra Pradesh, India. His academic and spiritual formation at Bharatiya Vidya Bhavans Public School prepared him well for his undergraduate work at Sri Chandrasekharendra Saraswathi Viswa Maha Vidyalaya, Enathur, Kanchipuram. He earned his Bachelor of Engineering in Electronics and Communications Engineering in 2003. During his undergraduate studies, he was involved in the development of a micro-controller based front panel for a Bit Error Rate Reader used in satellite receiver systems at National Remote Sensing Agency, Hyderabad, India.

Arvind will receive his master’s degree in electrical engineering from Louisiana State University, Baton Rouge, Louisiana, in 2006. His major area of concentration was communications/systems. He was involved in fund-raising efforts for various educational programs of the University. His last three semesters were spent working as a graduate assistant at the Louisiana Geological Survey. In his free time he enjoys playing guitar, swimming and reading.