A ROBUST IMAGE WATERMARKING SCHEME INVARIANT
TO ROTATION, SCALING AND TRANSLATION ATTACKS

By

Owalla Felix Ochieng

F56/76255/09

A thesis submitted in partial fulfillment for the degree of Master of Science in Electrical
and Electronic Engineering at the Department of Electrical and Information Engineering
in the University of Nairobi

July 28, 2014
DECLARATION OF ORIGINALITY FORM

Name of Student: Owalla Felix Ochieng
Registration Number: F56/76255/2009
College: College of Architecture and Engineering
Faculty/School/Institute: School of Engineering
Department: Department of Electrical and Information Engineering
Course Name: Master of Science in Electrical and Electronic Engineering
Title of the work: A Robust Image Watermarking Scheme Invariant to Rotation, Scaling and Translation Attacks

DECLARATION

1. I understand what Plagiarism is and I am aware of the University’s policy in this regard.
2. I declare that this thesis is my original work and has not been submitted elsewhere for examination, award of a degree or publication. Where other people’s work, or my own work has been used, this has properly been acknowledged and referenced in accordance with the University of Nairobi’s requirements.
3. I have not sought or used the services of any professional agencies to produce this work.
4. I have not allowed, and shall not allow anyone to copy my work with the intention of passing it off as his/her own work.
5. I understand that any false claim in respect of this work shall result in disciplinary action, in accordance with University Plagiarism Policy.

Signature…………………………………………………………………………

Date………………………………………………………………………………
APPROVAL

This thesis has been submitted for examination with my approval as university supervisor.

Professor Elijah Mwangi

Signature………………………………………………………………………

Date………………………………………………………………………
ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my supervisor Prof. Elijah Mwangi for his assistance during the research period and more so his role in securing the funding for the dissemination of the research. My thanks also goes to the Chief Executive Officer of the Kenya Education Network (KENET), Prof. Meoli Kashorda for the financial assistance that enabled me to participate in the IEEE MELECON and IEEE Africon conferences in order to get my work disseminated and published. I am also grateful to KENET’s Bernadette Mwandau, Ernest Kingi and Fernandes Korir for their assistance in the funding process. I am very grateful to the IEEE region 8 Voluntary Contribution Fund (VCF) for the grant to attend the IEEE Africon conference in Mauritius.

I am thankful to my colleagues Dickson Wambaa and Henry Kiragu for their suggestions that helped to improve the research work. Finally, I would like to thank my family Mom, Dad, Robert, Evans and Maureen for their financial support and encouragement during the entire course of my study.
ABSTRACT

Digital watermarking has been proposed as an effective means of copyright protection, copy protection and finger-printing of digital multimedia. These are all means of protecting the digital multimedia from illegal copying and/or distribution. The need for protection of digital multimedia is much more acute than their analogue counterparts. This is because they are transmitted over the internet exposing them to the risk of being copied illegally. Various methods of digital image watermarking have been proposed. Attackers always attempt to remove the embedded watermarks or render them useless by using either geometric attacks or signal processing attacks. Most watermarking algorithms are fairly robust to signal processing attacks but are fragile to geometric attacks.

This study proposes a watermarking scheme that is robust to geometric distortion attacks and signal processing attacks without increasing the computational complexity of the algorithm. The proposed method is based on embedding the watermark in the Discrete Cosine Transform (DCT) domain and then employing a spread-spectrum format to enhance security and Vector Quantization (VQ) techniques to compress the image. Since embedding the watermark in the DCT domain gives it sufficient robustness to signal processing attacks, the proposed technique focuses more on rotation, scaling and translation (RST) attacks. This method employs the Harris corner detector-based feature-points to obtain a Delaunay tessellation that is used in reversing the geometric attacks before attempting to extract the watermark.

In situations where the RST attacks lead to formation of substantially dark regions in the image, some feature-points are lost or additional feature-points are obtained leading to formation of a different Delaunay tessellation. This results in a watermark of poor quality or even entirely lost in some instances. A method of improving the estimation of the nature of attack and thereby leading to an improvement in the quality of the extracted watermark is proposed in this thesis. The method is based on the estimation of the RST attack of an image by taking a mean distortion on selected triangular regions.

The accuracy of the proposed method has been tested by computer simulation experiments using MATLAB. Various forms of attacks such as image filtering, image noising, histogram equalization, image cropping, JPEG compression and RST attacks on standard test images have been simulated and watermark recovery assessed. The test images that have been include the 256 × 256 8-bit grey level images ‘Pepper’, ‘Cameraman’ and ‘House’ and the 512 × 512 8-bit grey level images ‘Lifting body’, ‘Girl’ and ‘Baboon’ among others.

The proposed algorithm has also been tested on colour images such as ‘Baboon’, ‘Couple’ and ‘House’ and on video such as ‘Bus’, ‘Foreman’ and ‘Container’ among others. The Peak Signal-to-Noise Ratio (PSNR) and the Normalised Cross-correlation (NC) measures have been used to assess the quality of the watermarked images and recovered watermarks. The computer simulation results show that the proposed algorithm is effective in extraction of watermarks from images subjected to geometric attacks.
# ABBREVIATIONS AND ACRONYMS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDMA</td>
<td>Code Division Multiple Access</td>
</tr>
<tr>
<td>CIE</td>
<td>International Commission on Illumination</td>
</tr>
<tr>
<td>CMY</td>
<td>Cyan, Magenta and Yellow</td>
</tr>
<tr>
<td>CMYK</td>
<td>Cyan, Magenta, Yellow and Key</td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete Cosine Transform</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Fourier Transform</td>
</tr>
<tr>
<td>DVD</td>
<td>Digital Versatile Disk</td>
</tr>
<tr>
<td>DWT</td>
<td>Discrete Wavelet Transform</td>
</tr>
<tr>
<td>FMT</td>
<td>Fourier-Mellin Transform</td>
</tr>
<tr>
<td>FT</td>
<td>Fourier Transform</td>
</tr>
<tr>
<td>HSI</td>
<td>Hue, Saturation and Intensity</td>
</tr>
<tr>
<td>LBG</td>
<td>Linde, Buzo and Grey</td>
</tr>
<tr>
<td>LPM</td>
<td>Log-Polar Mapping</td>
</tr>
<tr>
<td>LSB</td>
<td>Least Significant Bit</td>
</tr>
<tr>
<td>LZW</td>
<td>Lempel-Ziv-Welch</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Square Error</td>
</tr>
<tr>
<td>MT</td>
<td>Mellin Transform</td>
</tr>
<tr>
<td>NC</td>
<td>Normalized Cross-correlation</td>
</tr>
<tr>
<td>NTSC</td>
<td>National Television System Committee</td>
</tr>
<tr>
<td>PSNR</td>
<td>Peak Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>RAM</td>
<td>Random Access Memory</td>
</tr>
<tr>
<td>RF</td>
<td>Rotation Factor</td>
</tr>
<tr>
<td>RGB</td>
<td>Red, Green and Blue</td>
</tr>
<tr>
<td>RST</td>
<td>Rotation, Scaling and Translation</td>
</tr>
<tr>
<td>SF</td>
<td>Scaling Factor</td>
</tr>
<tr>
<td>TF</td>
<td>Translation Factor</td>
</tr>
<tr>
<td>TV</td>
<td>Television</td>
</tr>
<tr>
<td>VLC</td>
<td>Variable Length Coding</td>
</tr>
<tr>
<td>VQ</td>
<td>Vector Quantization</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 3.1: The Voronoi tessellation of a set of points and the corresponding Delaunay tessellation ................................................................. 23
Figure 3.2: Delaunay triangulations of the feature points for Baboon and Boat test images ................................................................. 23
Figure 3.3: Definition of DCT regions ................................................................. 25
Figure 3.4: Zigzag re-ordering of the DCT coefficients ........................................... 28
Figure 3.5: Block diagram on VQ encoding and decoding processes ......................... 30
Figure 5.1: The Chromaticity diagram ................................................................. 42
Figure 5.2: The primary colours triangle in the chromaticity diagram ....................... 42
Figure 5.3: The RGB cube .................................................................................. 43
Figure 5.4: The Peppers image in RGB and converted to CMY colour space .......... 44
Figure 5.5: Peppers image in RGB colour space and converted to NTSC ............... 46
Figure 5.6: Peppers image in RGB colour space and converted to YCbCr .......... 47
Figure 5.7: Peppers image in RGB and HSV colour space ..................................... 48
Figure 6.1: Cover image and watermarked image at PSNR of 34.9 dB .................. 60
Figure 6.2: Cover image and Watermarked image at PSNR of 38.7 dB ............... 60
Figure 6.3: JPEG Compression attacked Cameraman image with a Quality Factor of 50%, the embedded watermark and the retrieved watermark .......... 61
Figure 6.4: JPEG Compression attacked Lena image with a Quality Factor of 50%, the embedded watermark and the retrieved watermark .................... 62
Figure 6.5: JPEG Compression attacked images with a Quality Factor of 30%, and the retrieved watermark .......................................................... 62
Figure 6.6: Histogram Equalisation attacked images and the extracted watermark .... 63
Figure 6.7: Extracted watermarks from images subjected to various Geometric Distortion attacks ................................................................. 64
Figure 6.8: Delaunay triangulations used in the synchronisation process ............... 65
Figure 6.9: Probe triangle used in estimating angle of rotation of Cameraman image as proposed by Tang and the extracted watermark .................. 65
Figure 6.10: Probe triangle used in estimating angle of rotation of Lena image as proposed by Tang and the extracted watermark ......................... 66
Figure 6.11: Probe triangles used in the proposed technique to estimate the angle of rotation of Cameraman image and the extracted watermark ............................... 66
Figure 6.12: Probe triangles used in the proposed technique to estimate the angle of rotation of Lena image and the extracted watermark ............................... 67
Figure 6.13: Plot showing comparison of quality of extracted message between Tang’s, Pal’s and the proposed techniques after the watermarked Cameraman image is subjected to various angles of rotation attack ................................. 68
Figure 6.14: Plots showing comparison of quality of extracted message between Tang’s, Pal’s and the proposed techniques after the watermarked Lena image is subjected to various angles of rotation attack .......................................... 68
Figure 6.15: A chart showing the graphical comparison of the quality of watermark extracted from Baboon image after it was subjected to various attacks .......... 69
Figure 6.16: Red, Green and Blue component images of the Peppers and the corresponding extracted watermarks ......................................................... 71
Figure 6.17: Red, Green and Blue component images of the Baboon and the corresponding extracted watermarks ......................................................... 72
Figure 6.18: The Couple and House images subjected to Gaussian noise attack with a mean of zero and variance of 0.02, and the extracted watermark ...................... 73
Figure 6.19: Delaunay triangulation of the cover image and the attacked Girl image ....... 74
Figure 6.20: Delaunay triangulation of the cover image and the attacked House image .... 74
Figure 6.21: Target and probe triangles of Girl image at RF of 39° and the extracted watermark ......................................................................................... 75
Figure 6.22: Target and probe triangles of House image at RF of 39° and the extracted watermark ......................................................................................... 75
Figure 6.23: Cyan, Magenta and Yellow components of the Peppers image ............... 77
Figure 6.24: Luminance, Hue and Saturation components of Peppers image and the corresponding watermark extracted from each ................................. 78
Figure 6.25: Luminance, Hue and Saturation components of Baboon image and the corresponding watermark extracted from each ................................. 78
Figure 6.26: The probe and target triangles of the Peppers in NTSC colour space and the retrieved watermark ......................................................................................... 79
Figure 6.27: The probe and target triangles of the Baboon in NTSC colour space and the retrieved watermark ................................................................. 80
Figure 6.28: Hue, Saturation and Value component images of the House and the corresponding watermark extracted from each ........................................ 80
Figure 6.29: Hue, Saturation and Value component images of the Girl and the corresponding watermark extracted from each ................................. 81
Figure 6.30: Component images of the House in YCbCr colour space and the corresponding watermark extracted from each ................................. 81
Figure 6.31: Component images of the Girl in YCbCr colour space and the corresponding watermark extracted from each ........................................ 82
Figure 6.32: Cover and watermarked frames of the Foreman and the extracted message ................................................................. 83
Figure 6.33: Cover and watermarked frames of the Bus video and the extracted message ................................................................. 83
Figure 6.34: Cover and watermarked frames of the Container test video and the extracted message ................................................................. 84
Figure B.1: Homepage of the IEEE MELECON 2012 ........................................ 104
Figure C.1 Homepage of the IEEE Africon 2013 ........................................ 110
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Quantisation values of JPEG compression scheme</td>
<td>27</td>
</tr>
<tr>
<td>6.1</td>
<td>A comparison of the performance of the proposed watermarking technique with some related techniques</td>
<td>70</td>
</tr>
<tr>
<td>6.2</td>
<td>Summary of performance of the proposed technique to affine geometric attacks on RGB colour images</td>
<td>76</td>
</tr>
<tr>
<td>6.3</td>
<td>Comparing the performances of video watermarking techniques</td>
<td>84</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

DECLARATION OF ORIGINALITY FORM ................................................................. ii
APPROVAL ............................................................................................................. iii
ACKNOWLEDGEMENTS ....................................................................................... iv
ABSTRACT ............................................................................................................ v
ABBREVIATIONS AND ACRONYMS .................................................................. vi
LIST OF FIGURES ................................................................................................... vii
LIST OF TABLES ...................................................................................................... x

CHAPTER ONE: INTRODUCTION .......................................................................... 1
  1.1. Background to the Study .............................................................................. 1
  1.2. Properties of a Digital Image Watermark ................................................... 2
  1.3. Applications of Digital Image Watermarking .............................................. 4
  1.4. Watermark Embedding ................................................................................ 5
  1.5. Problem statement ....................................................................................... 6
  1.6. Objectives ................................................................................................... 7
  1.7. Scope of Work ............................................................................................. 7
  1.8. Thesis Organisation ..................................................................................... 7

CHAPTER TWO: LITERATURE REVIEW .............................................................. 9
  2.1. Spatial Domain Watermarking Techniques ................................................ 9
  2.1.1. Visible and Invisible Watermarks ........................................................... 9
  2.1.2. LSB modification method ...................................................................... 10
  2.1.3. CDMA Techniques ................................................................................ 10
  2.2. Transform Domain Watermarking Techniques ......................................... 11
  2.2.1. Discrete Fourier Transform (DFT) Domain ........................................... 12
  2.2.2. Fourier- Mellin Transform ................................................................... 13
  2.2.3. Discrete Cosine Transform (DCT) Domain .......................................... 14
  2.2.4. Discrete Wavelet Transform (DWT) Domain ....................................... 15
  2.3. Invariance to Geometric Attacks ................................................................ 16
  2.3.1. Invariant domain-based techniques ...................................................... 17
  2.3.2. Synchronisation-based techniques ......................................................... 17
  2.3.3. Feature-Point-based techniques ............................................................. 18
  2.4. Proposed Technique ................................................................................. 18

CHAPTER THREE: IMAGE TESSELLATION AND DCT/VQ TECHNIQUES .......... 20
  3.1. Feature-point detection ............................................................................. 20
  3.1.1. Harris Corner Detector ......................................................................... 21
3.2. Voronoi and Delaunay Tessellations .......................................................... 22
3.3. Discrete Cosine Transform ........................................................................ 24
3.4. Vector Quantization (VQ) Theory .............................................................. 28

CHAPTER FOUR: GREY-SCALE IMAGE WATERMARKING ................................32
4.1. Motivation ................................................................................................... 32
4.2. Image Format .............................................................................................. 32
4.3. Spatial domain watermarking techniques ................................................... 33
   4.3.1. Visible watermarking ........................................................................... 34
   4.3.2. LSB watermarking .............................................................................. 34
   4.3.3. CDMA watermarking .......................................................................... 35
4.4. Transform domain watermarking techniques ............................................ 35
   4.4.1. Fourier-Mellin Transform watermarking ............................................. 36
   4.4.2. Discrete Wavelet Transform watermarking ........................................ 36
   4.4.3. Proposed watermarking technique ..................................................... 37
4.5. Watermark Extraction ................................................................................ 38
4.6. Note on Publication .................................................................................... 39

CHAPTER FIVE: COLOUR IMAGE AND VIDEO WATERMARKING .............40
5.1. Motivation ................................................................................................... 40
5.2. Basics of Chromaticity .............................................................................. 40
5.3. Colour Spaces ............................................................................................ 43
   5.3.1. RGB colour space .............................................................................. 43
   5.3.2. CMY and CMYK colour space ........................................................... 44
   5.3.3. National Television System Committee (NTSC) colour space .......... 45
   5.3.4. YCbCr colour space ........................................................................... 46
   5.3.5. HSV colour space ............................................................................. 47
   5.3.6. HSI colour space .............................................................................. 49
5.4. Video Formats ........................................................................................... 49
5.5. Feature-Points Detection and Tessellation in Colour Images .................... 50
5.6. VQ and DCT in Colour Images ................................................................ 51
   5.6.1. VQ in colour images ......................................................................... 51
   5.6.2. DCT in colour images ....................................................................... 52
5.7. Watermark Embedding in Colour Images ................................................ 52
5.8. Watermark Extraction in Colour Images .................................................. 53
5.9. Watermark Embedding and Extraction in Videos ..................................... 54
5.10. Note on Publication .................................................................................. 56
CHAPTER SIX: RESULTS AND DISCUSSIONS .................................................................57
  6.1. Quality Measures ..........................................................................................57
  6.2. Computer Simulation Results ......................................................................58
  6.3. Grey-scale image watermarking results .....................................................59
  6.4. Colour image watermarking results ............................................................71
  6.5. Video watermarking ....................................................................................82
CHAPTER SEVEN: CONCLUSION AND RECOMMENDATIONS ..........................85
  7.1. Conclusions ..................................................................................................85
  7.2. Recommendations .......................................................................................86
REFERENCES ..........................................................................................................88
APPENDIX A: MATLAB PROGRAMS .................................................................94
APPENDIX B: PUBLICATION 1 .............................................................................104
APPENDIX C: PUBLICATION 2 .............................................................................110
CHAPTER 1
INTRODUCTION

1.1. Background to the Study
There has been significant progress in digital multimedia technology with the advancement of internet connectivity. The World Wide Web and its associated technologies have led to the ease of transmission of digital multimedia, editing of digital content and also duplication of digital content without loss of quality. Unfortunately these advancements have been accompanied by illegal production, distribution and even counterfeiting of digital multimedia.

Several solutions have been suggested to eradicate these problems. The solutions have been a combination of registration such as the enforcement of Digital Rights Management (DRM) and through technologies such as the inclusion of cryptography and watermarking techniques. In the DRM system, the access is controlled by the hardware such that the multimedia can only be accessed through specific hardware. The DRM scheme has several disadvantages as its mode of restriction prevents users from doing some legal operations such as making back-up copies of CDs or DVDs of which evidence can be deduced and employed in a legal process. Cryptography involves the use of secret code to scramble a file into a format that is unusable by unauthorized users and can also be used in the protection of digital product from illegal access. However once the digital product is decrypted it is no longer protected and can be accessed by anyone. The art of digital watermarking involves embedding of a logo or signature in the product for authentication or verification purposes. Though it has vulnerabilities, it has emerged to be superior as it can be used to accomplish a wide variety of functions and can be used to complement the other techniques [1].
Chapter 1: Introduction

A closely related technique to watermarking is steganography which is the art of hiding a secret message in a product. The aim is to communicate a message rather than protect the product. The work in this thesis is concerned with the watermarking of digital images and hence the issues of DRM, cryptography and steganography shall not be pursued further.

1.2. Properties of a Digital Image Watermark

The specific image that is to be watermarked is referred to as the cover image, the secret signature or logo that is to be embedded in the cover image is termed the watermark. The watermark is embedded into the cover image using a secret key which then gives a watermarked image as the end product. Only the owner and authorised users have access to the key and it is not possible to remove the watermark without the knowledge of the key. Watermarking is then defined as the practice of imperceptibly altering a cover image to embed a message on the ownership or authenticity [2].

A good watermark should exhibit the following properties.

1.2.1. Robustness

A watermarking algorithm is said to be robust if the embedded watermark can be retrieved effectively with minimal distortion after the watermarked image has been subjected to attacks. There are various forms of watermark attacks that are used to render the watermark useless or remove it entirely from the image. The most common ones are classified in two categories: signal processing attacks and geometric distortion attacks [3]. The signal processing attacks are the attacks that affect the watermark during signal processing functions such as filtering, compression and denoising. The geometric distortion attacks are the attacks that occur when the cover image is geometrically transformed or manipulated such as rotation, scaling and translation. Of these forms of attack, geometric attacks are the hardest to deal with since they
lead to synchronisation errors. There are several techniques often applied to recover from geometric attacks in watermarked images. These are template-based, moment-based, invariant domain-based and feature-points-based techniques which are discussed in later sections.

1.2.2. Embedding Effectiveness
Embedding effectiveness is the probability that the output of the embedder will be watermarked. In other words, it is the probability that the embedded watermark can be detected right after the embedding process. The desired embedding effectiveness of unity is however not always attainable.

1.2.3. Fidelity
Fidelity is the perceptual similarity between the original work and the watermarked work. Watermarks can be classified in terms of fidelity into visible and invisible watermarks. In invisible watermarking, the watermarked image should look similar to the original image without any perceivable distortions.

1.2.4. Complexity
Complexity refers to the difficulty of the embedding and extracting algorithm. In some cases the cost of hardware used in the watermarking process is also considered under the complexity. The algorithm should be simple for the authorized user and very complex to any unauthorized users.

1.2.5. Data Payload
Data payload is the capacity of information that can be hidden into the work without causing significant loss in quality of the cover work. In most watermarking algorithms, there is usually a trade-off between robustness, fidelity and capacity.
1.2.6. Blind or Informed Detection

Blind detection is the ability of the detecting algorithm to extract the embedded watermark without reference to the original unwatermarked work. Informed detection however, requires the original work in order to successfully detect the watermark.

1.2.7. False Positive Rate

A false positive is the detection of a watermark in a work that does not contain any watermark. The false positive rate refers to the number of false positives we expect to occur in a given number of runs of the detector.

1.3. Applications of Digital Image Watermarking

Of the many applications of digital image watermarking, a few are discussed here.

i. Ownership Identification

Watermarks can be used to identify the owner of some multimedia content. By embedding the owner’s logo, the watermark can be used to assert ownership of the multimedia if dispute were to arise.

ii. Transaction Tracking

Watermarks can be used in tracking legitimate users of digital media content in a process commonly referred to as fingerprinting. This process can also be used in tracing the origin of illegal copies of the digital content [2].

iii. Content Authentication

In cases where an image or video has been tampered with, a fragile watermark can be used to detect editing of the content. This is mostly used when the content is used as evidence in court of law [1].
iv. Broadcast Monitoring
Watermarks can also be used to monitor if certain multimedia content has been aired. This is mostly used by advertisers to monitor if TV stations air their advertisements as paid for. It is also used in Radio Data Systems (RDS) so that when certain contents are being aired then the radio automatically tunes to that frequency [2].

1.4. Watermark Embedding
There are several methods of embedding a watermark signal into the cover image. The techniques can be divided into either the spatial domain or the transform domain. The spatial domain techniques mainly comprise of the Least Significant Bit (LSB) modification and spread-spectrum techniques. The spatial domain techniques are usually simple to implement and have a very high capacity but they suffer from very low robustness levels. The transform domain techniques comprise of the use of Discrete Fourier Transform (DFT), Discrete Wavelet Transform (DWT), Fourier-Mellin Transform (FMT) and Discrete Cosine Transform (DCT). The frequency or transform domain techniques are more robust than their spatial domain counterparts. However they do not have the capacity of the spatial domain techniques and are more complex to implement. They are suitable for watermarking since watermarks require robustness and high capacity is not necessary [1], [2]. There are variations of the transform domain techniques where the watermark is embedded in the coefficients of a codebook obtained from coding the image through a Vector Quantisation (VQ) process. This variation has been reported to improve the robustness of the watermark in addition to compressing the image thereby improving transmission and storage efficiency.

Generally, the watermark is embedded using either of the following techniques. The first technique is done by adding a strength-scaled watermark into the work as given by equation
Chapter 1: Introduction

(1.1) This technique can be used in both spatial and transform domain watermarking. The second technique is done by interchanging coefficients of the transformed work in a certain pattern and can only be done in the transform domain watermarking.

\[ I^w(u, v) = I(u, v) + \beta W(u, v) \quad (1.1) \]

Where \( I^w(u,v) \) is the watermarked image, \( I(u,v) \) is the cover image, \( W(u,v) \) is the watermark and \( \beta \) the embedding or scaling strength.

1.5. Problem statement

Watermarking is one of the most widely used methods of copyright protection and fingerprinting of images, however most of the techniques are susceptible to a variety of attacks. Since robustness of a watermark is achieved via computational complexity and image quality trade-offs, a watermark is therefore considered robust when an attacker would find it computationally expensive to remove the watermark or severely damage the quality of the image in the process of removing it. The main challenge with digital image watermarking is that most watermarking techniques are fragile to either signal processing attacks or geometric attacks. Also in order to efficiently store or transmit images, they need to be compressed, however watermarking involves adding information to the image, and this leads to inefficiency in the storage or transmission.

The investigation in the research work presented in this thesis exploits VQ technologies to compress the image and the DCT to split the encoded image into perceptual bands in order to attain robustness to signal processing attacks. Harris corner detector-based feature-points are used to obtain synchronisation of a geometrically attacked image. The trade-off between perceptual quality and robustness of a watermark is achieved by exploiting those techniques.
Chapter 1: Introduction

1.6. Objectives

i. Main Objectives

To obtain a visually imperceptible watermark by using VQ, Harris corner detector-based feature-points, Delaunay tessellation and DCT that is robust to signal processing and RST attacks. In addition the watermarked image should have no significant fidelity distortion and must be visually indistinguishable from the original image.

ii. Specific Objectives

To improve on the robustness of the watermark by improving the quality of the extracted watermark after the image has been subjected to a variety of signal processing and geometric distortion attacks.

1.7. Scope of Work

The scope of this study shall entail image compression using VQ, embedding and retrieving watermark in the DCT domain and synchronizing attacked images so as to be able to retrieve the embedded watermarks successfully even after RST attacks on the image by using Harris corner detector-based feature-points and Delaunay tessellations.

1.8. Thesis Organisation

This thesis is organised as follows. This first chapter presents an introduction to image watermarking and the objectives of the research work that is reported. In chapter 2, the literature review that reports on various watermarking techniques that have been proposed by other researchers is presented. Chapter 3 covers feature-points extraction, image tessellation and DCT
Chapter 1: Introduction

and VQ techniques. Chapter 4 covers the grey-scale image watermarking process where the proposed technique is presented while chapter 5 covers colour and video watermarking process. The results and discussions are covered in chapter 6 and lastly chapter 7 gives the conclusion and recommendations.
CHAPTER 2
LITERATURE REVIEW

This chapter gives a brief review of the techniques that have been proposed or used in digital image watermarking. It reviews the main spatial and transform domain watermarking techniques, invariance to geometric distortion attacks and the proposed technique.

2.1. Spatial Domain Watermarking Techniques

In spatial domain watermarking, the message is embedded into the cover image by altering the intensities of the host image pixels. These techniques generally have a very high capacity and are simple to implement compared to their transform domain counterparts [1]. However they have the disadvantage of fragility to most attacks hence they are suitable for applications that require capacity but not robustness. The spatial domain techniques can be broadly classified into visible and invisible watermarks.

2.1.1. Visible and Invisible Watermarks

Visible watermarking, as the name suggests, involves embedding the watermark in such a way that it is perceptible to the human observer. The technique combines the watermark and the cover image by modifying the brightness of the cover image as a function of the watermark and a secret key [1]. This technique has been extensively used by the IBM digital libraries project. It however has a major limitation of being easily attacked due to its visible nature. In addition it is susceptible to averaging and image in-painting attacks [4]. For these reasons, visible watermarking techniques have not been considered in the work presented in this thesis.
2.1.2. LSB modification method

The Least Significant Bit (LSB) modification is implemented by modifying the least significant bit of the cover-image in the areas where the watermark is to be embedded. An attacker can make all the pixels to have either an even or an odd intensity level thereby altering the watermarked and non-watermarked pixels of the image hence effectively removing the watermark without affecting the quality of the image. This technique is also very fragile to signal processing attacks such as filtering and compression. In attempts to make the LSB modification process more robust, spread-spectrum technique is used to hide the positions of the message being embedded [1].

Li et al [5] proposed expansion of prediction-error in the coding process to increase the capacity of the watermarking algorithm. They also proposed selection of pixels to be embedded with the watermark so as to minimise the degradation of the quality of the watermarked image. However their technique does not address the issue of fragility to most of the attacks that is characteristic to the LSB modification technique of watermarking.

2.1.3. CDMA Techniques

The Code Division Multiple Access (CDMA) methods find their applications in both spatial and transform domain image watermarking techniques. In the spatial domain, the CDMA technique is used to improve the robustness of the LSB technique by using Pseudo-random Noise (PN) to determine the pixels used for embedding the watermark based on a given key seeded to the pseudo-random number generator. As proposed by Katzenbeisser and Petitcolas [1], exploiting the correlation properties of a PN pattern to add the watermark to the cover image can be used to
Chapter 2: Literature Review

hide the pixels chosen for embedding. To extract the watermark, the pseudo-random number generator is seeded with the same key and a correlation measure between the pattern and the watermarked image is computed. If the correlation exceeds a certain threshold the watermark is detected and a single bit is set. In a multiple bit watermark, it is extended to the multiple bits by dividing the image into blocks as proposed by Kutter et al [6].

Cox et al [7] proposed that the watermark could be spread to all the coefficients of a transformed cover image using a PN sequence. The same was adopted and applied to the spatial domain by Langelaar et al [8] where a PN sequence is generated for each value of the watermark using an independent key. The summation of these PN sequences was then used to represent the watermark and is scaled and added to the cover image. To detect the watermark, each key is used to generate its PN sequence which is then correlated to the entire image.

Though the CDMA techniques have improved the robustness of the spatial domain techniques of watermarking in terms of the embedding pixels, they are still found to lack robustness to most signal processing attacks as compared to the transform domain techniques.

2.2. Transform Domain Watermarking Techniques

The spatial domain watermarking techniques are computationally simple to implement, however they are very fragile to most forms of attacks. The robustness and quality of a watermark can be improved by exploiting the properties of the cover image during the embedding process. Transform domain watermarking techniques give the best platform to exploit these properties by splitting the cover image into frequency bands and therefore allowing for a better trade-off between the robustness and quality of the embedded watermark. The mid-frequency bands are found to give the best trade-off as they are perceptually invisible and less prone to compression
attacks [1] and [2]. The following sections deal with the commonly used transform domain techniques.

2.2.1. Discrete Fourier Transform (DFT) Domain

The Discrete Fourier Transform (DFT) is widely used in signal processing and was adopted to image watermarking as it gave the possibility of manipulating the frequencies of the cover image. It is important to select frequencies of the cover image which then can be used in the embedding process in order to give the best trade-off between perceptibility and robustness.

The 2 dimensional DFT of an $M \times N$ image is obtained using the following equation [1].

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j\pi \left( \frac{ux}{M} + \frac{vy}{N} \right)}$$

for $x = 0, 1, 2, \ldots, M - 1$

$y = 0, 1, 2, \ldots, N - 1$ \hspace{1cm} (2.1)

where $f(x,y)$ is the grey-scale level of the pixel at position $(x,y)$ in the image and $F(u,v)$ is its DFT coefficient.

In order to obtain the spatial image from the DFT coefficients an inverse DFT (IDFT) of the coefficients is performed using the following equation.

$$f(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(u, v) e^{j\pi \left( \frac{ux}{M} + \frac{vy}{N} \right)}$$

for $u = 0, 1, 2, \ldots, M - 1$

$v = 0, 1, 2, \ldots, N - 1$ \hspace{1cm} (2.2)

A derived form of DFT known as the Fourier-Mellin Transform is more commonly used in digital image watermarking [1].
Chapter 2: Literature Review

2.2.2. Fourier- Mellin Transform

The Fourier-Mellin transform has been extensively used in image recognition because of its invariance to rotation, translation and scale. Due to its invariance to rotation, translation and scale the Fourier-Mellin Transform can also be used in detection of watermarks which have been subjected to most geometric attacks [9], [10] and [11].

To obtain the Fourier-Mellin Transform of an image \( I(x,y) \), the coordinates of the image are first converted from Cartesian to Polar coordinates given by \( I(r,\theta) \) in a process known as Log-Polar Mapping (LPM). The Fourier transform of the polar coordinates are then obtained as defined in equation (2.3) [12].

\[
I_f(k, v) = \frac{1}{2\pi} \int_0^\infty \int_0^{2\pi} I(r,\theta) r^{-jv} e^{-ik\theta} d\theta \frac{dr}{r}
\]

(2.3)

Where \( r = \sqrt{x^2 + y^2} \)

and \( \theta = \tan^{-1}\left(\frac{y}{x}\right) \)

\( \forall x, y > 0 \)

And the inverse is obtained by:

\[
I(r,\theta) = \int_{-\infty}^{+\infty} \sum_{k>0} I_f(k, v)r^{jv} e^{jk\theta} dv
\]

(2.4)

Lin et al [10] and Kim et al [11] proposed the Log-Polar Mapping of the DFT coefficients of the cover image before embedding the watermark. In their proposal they exploit the shift-invariant property of DFT which gives coefficients which are translation invariant. The LPM of every magnitude of the DFT coefficients is then determined to obtain a rotation and scale invariant domain. The watermark is then embedded in the log-polar coordinates making it invariant to rotation, scaling and translation attacks.
Chapter 2: Literature Review

Fourier-Mellin Transform is computationally complex as it requires the derivation of the LPM on DFT domain, a process that requires interpolation of DFT magnitudes with large dynamic range between neighbouring coefficients. In addition the DFT process involves dealing with complex numbers further complicating the computation process.

2.2.3. Discrete Cosine Transform (DCT) Domain

This is the most commonly used transform domain watermarking technique due to its energy compaction properties and ease of computation [1] and [13].

This thesis focuses on this domain of watermarking and the improvements that can be made to improve its robustness. Lin and Chin [14] proposed embedding the watermark in the mid-frequency DCT coefficients of the cover image in order to attain sufficient trade-off between robustness to signal processing attacks and perceptual quality of the watermarked image.

Shoemaker [15] proposed a DCT domain embedding technique that relies on the JPEG quantisation, where the coefficients which have equal quantisation level are compared and exchanged in such a way to signify if a bit has been embedded in them or not. In this technique, before the DCT coefficients are computed, the image is decomposed into non-overlapping blocks of size 8×8 pixels. The mid-frequency bands of DCT blocks are then used in the embedding process in spread-spectrum format to improve on robustness. This technique is robust to signal processing attacks but is not robust to geometric attacks.

Huang et al, in their work [16] proposed watermarking with VQ block means and variances and embedding using grey-level watermarks. Pal et al [17] propose the use of a modified LBG algorithm in the design of the codebook. After encoding the image, the codewords within the codebook are shuffled to improve confidentiality of the watermark. The DCT of the shuffled codebook is obtained and the watermark is embedded in the mid-band frequencies of the shuffled
DCT coefficients. These VQ techniques exhibit an improved robustness to some geometric attacks such as cropping and small angles of rotation.

A combination of Singular Value Decomposition (SVD) and DCT was proposed by Rafzul [18]. This technique is reported to attain resilience to signal processing attacks and some geometric distortion attacks. The technique, however does not give satisfactory results for an image that has been subjected to rotation and translation attacks.

The DCT domain techniques of watermarking exhibit sufficient robustness to signal processing attacks but are inadequate as far as geometric attacks are concerned.

2.2.4. Discrete Wavelet Transform (DWT) Domain

The DWT consists of multi-scale spatial-frequency decomposition of a signal into low frequency approximation image, horizontal, vertical and diagonal details. Two-channel filter banks are used in obtaining these different levels of resolution and are defined by the equations:

\[
\psi^H(x, y) = \psi(x)\varphi(y) \\
\psi^V(x, y) = \varphi(x)\psi(y) \\
\psi^D(x, y) = \psi(x)\psi(y)
\]

(2.5)

Where

\[
\psi(x) = \sum_n h_{\psi}(n)\sqrt{2}\varphi(2x - n) \\
\varphi(x) = \sum_n h_{\varphi}(n)\sqrt{2}\varphi(2x - n)
\]

(2.6)

Where \(h_{\psi}\) and \(h_{\varphi}\) represent the scaling and wavelet vectors respectively.
Chapter 2: Literature Review

When applied to images, it is known that the DWT domain models the Human Visual System more closely than the DCT, however embedding the watermark in the DWT domain is computationally more complex than in the DCT domain. It has also been reported that if encoders of the same level of technology were used in both the DCT and DWT, the difference in their performance will be insignificant while the computational cost and complexity increases significantly [19].

Al-Haj [20] and Lai and Tsai [21] have reported that in comparison to DCT domain techniques, DWT domain slightly improves the perceptual quality of the watermarked image but still has insufficient robustness to geometric attacks. Al-Haj proposes a combination of DWT and DCT to further improve the perceptual quality and robustness to geometric attacks. He proposes a DCT decomposition of the horizontal and diagonal details of a DWT transformed image before embedding the watermark. This technique however fails to address the issue of geometric attacks. Lai and Tsai propose use of DWT and SVD to improve robustness to signal processing and geometric attacks. Their proposal exhibits improved robustness to numerous attacks but at the same time increasing computational complexity of the embedding and extraction processes.

2.3. Invariance to Geometric Attacks

Transform domain techniques have shown an improved robustness to signal processing attacks, they however lack robustness to geometric processing attack. Geometric attacks-focused watermarking techniques can be classified into invariant domain-based, synchronization-based and feature-point-based techniques. The following subsections give a brief discussion on these three techniques.
2.3.1. Invariant Domain-based techniques

The invariant domain-based techniques involve embedding the watermark in a domain that is invariant to the geometric attacks and they include FMT and SVD. Proposals have been made to embed the watermark in the FMT domain by various authors [9], [10], [11], [12]. However this technique poses the challenge of increasing the computational complexity of the embedding process and in addition it has been reported to be fragile to cropping attacks. The methods proposed in [18] and [21] employ a combination of SVD and DCT and also SVD with DWT to realize a solution to geometric attack invariance. These techniques improve the robustness of DCT and DWT to geometric attacks but with a substantial increment in computational complexity in the embedding and extraction processes.

2.3.2. Synchronisation-based techniques

Some synchronisation-based techniques involve embedding of a template in an image while others employ invariant moments to restore a geometrically attacked image back to its original position before the watermark is extracted. Pereira et al [22], [23] proposed a technique where log-polar maps are used to embed a geometric-attack-invariant template before the watermark is embedded in the Fourier Transform domain.

Foo and Dong [24] proposed Zernike moments obtained from a normalised image and then employed to determine the distortion by comparing the attacked image to the normalised image. In addition to being computationally complex, these techniques are found to increase the data overhead in the cover image leading to inefficiency during storage or transmission of the watermarked image.
2.3.3. Feature-Point-based techniques

These techniques involve extraction of feature-points which are employed in restoring a geometrically transformed image back to its original position. They are more computationally efficient compared to synchronisation and invariant-domain based techniques and hence more popular.

In the work by Qi [25] and Tang and Hang [26] techniques that employ the extracted feature-points to get a Delaunay tessellation which is then used to reverse the geometric attack an image has been subjected to before attempting the recovery of the watermark have been proposed. These techniques are sufficiently robust to small magnitudes of attacks but they lose robustness as the magnitude increases as some feature-points are lost. In addition these techniques are found to be very fragile to cropping attacks on the watermarked image.

Lowe [27] proposed a method of extracting feature-points which used cascade filtering approach. The technique could extract feature-points even after the image has been subjected to large scaling magnitudes. This technique however, does not address large magnitudes of rotation and translation attacks and exhibits increased computational complexity with each successive filter.

2.4. Proposed Technique

This thesis proposes a technique that exploits the qualities of the VQ-DCT techniques proposed by Huang et al [16] and Pal et al [17] and those of feature-point based watermarking proposed by Qi and Qi [25] and Tang and Hang [26]. In the VQ-DCT techniques the watermark is embedded in a spread-spectrum format in the DCT coefficients of the VQ encoded codebook of the cover image and then the codebook is decoded to give the watermarked image.
Chapter 2: Literature Review

The VQ-DCT techniques are sufficiently robust to signal processing and cropping attacks while the feature-point based techniques are robust to rotation, scaling and translation attacks up to a certain magnitude.

When the angles of rotation or magnitude of scaling or translation result into formation of substantially large dark regions in the image, it leads to loss of some feature-points. The loss of feature-points leads to formation of a Delaunay tessellation that is different from the tessellation obtained when the image is at its original position hence restoring such an attacked image back to its original position becomes difficult. In order to restore such an image back to its original position, an averaging of distortions attained by several triangles in the Delaunay tessellation give a better estimate of the nature of the geometric attack hence leading to a more accurate restoration and better quality of the extracted watermark [28].

19
CHAPTER 3

IMAGE TESSELLATION AND DCT/VQ TECHNIQUES

In this chapter a review of the VQ theory and the DCT techniques and how they are used in the watermark embedding process is discussed. Feature-point extraction and the Delaunay triangulation of the feature-points are also discussed and how they are used in synchronizing geometrically distorted images before extracting the watermark is presented.

3.1. Feature-point detection

Feature-points are distinguishing characteristics or attributes on an image. Some features are natural, in that they form the visual appearance of an image, while others are as a result of manipulations of the image hence they are artificial. Image feature-points have found applications in image segmentation, image classification, image registration, and image synchronization [6], [25], [26].

There are various methods used in extracting feature-points from an image, of which the most commonly used are based on the Mexican hat wavelet and the Harris corner detector. The properties of wavelets are exploited while applying Mexican hat wavelet feature-point detection to detect features at different scales. This makes it unsuitable for detection at very fine scales and also in cases of local distortion of images [25]. The Harris corner detector is chosen for further investigation as it is reported to be robust to both global and local attacks whereas Mexican hat wavelet is fragile to local distortions.
Chapter 3: Image Tessellation and DCT/VQ Techniques

3.1.1. Harris Corner Detector
Harris and Stephens [29] improved on an earlier auto-correlation based corner detector technique which measured local changes on an image when a window is shifted by small amounts. The auto-correlation detector was found to be resilient to signal processing attacks such as illumination variation and also robust to geometric attacks such as rotation, scaling and translation. However, it could only detect discrete shifts at 45° angles and gave a noisy output because of the use of a rectangular window. The Harris corner detector employ the use of an auto-correlation function with a Gaussian window which is circular hence solving the problems of limited angles of detection and noisy outputs.

If the windowed region of the image is shifted by $(\Delta x, \Delta y)$ around a point $(x, y)$, the auto-correlation function is defined as

$$c(x, y) = \sum_w [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2$$

(3.1)

where $I(\ldots)$ is the image function and $(x_i,y_i)$ are points in the Gaussian window $W$ centered on $(x,y)$.

The shifted image is approximated by a Taylor’s expansion truncated to the first order terms as given by equation (3.2).

$$I(x_i + \Delta x, y_i + \Delta y) \approx I(x_i, y_i) + [I_x(x_i, y_i) I_y(x_i, y_i)]^{\Delta x \Delta y}$$

(3.2)

Equation (3.2) is then substituted into equ. (3.1) and then derived to give equ. (3.3)

$$c(x, y) = (\Delta x \ \Delta y) C(x, y)$$

(3.3)

where matrix $C(x,y)$ is given by

$$C(x, y) = \begin{bmatrix} \sum_w (I_x(x_i, y_i))^2 & \sum_w I_x(x_i, y_i)I_y(x_i, y_i) \\ \sum_w I_x(x_i, y_i)I_y(x_i, y_i) & \sum_w (I_y(x_i, y_i))^2 \end{bmatrix}$$

(3.4)
Chapter 3: Image Tessellation and DCT/VQ Techniques

where $I_x(x_i,y_i)$ and $I_y(x_i,y_i)$ are partial derivatives of the Image function $I(x_i,y_i)$ in $x$ and $y$, respectively [29].

The auto-correlation detector uses the eigenvalues $\lambda_1$, $\lambda_2$ of the matrix $C(x,y)$ to consider the features that can be obtained in the selected window. Where both the eigenvalues have small value, this indicates there is no feature of interest in the windowed image. If one eigenvalue is high while the other is low it is an indication of an edge in the windowed image. Finally, if both eigenvalues are high it indicates a corner in the windowed image [26].

In order to improve the response and robustness of the auto-correlation detector, Harris and Stephens employed the use of the trace and determinant rather than the eigenvalues of the matrix to determine the nature of the windowed patch. The response $R$ is obtained as shown in equation (3.5) [29].

$$R = \text{Det} - (k \times \text{Tr}^2)$$

where $\text{Det}$ and $\text{Tr}$ are the determinant and trace of matrix $C(x,y)$ respectively and $k$ is the factor used to set the threshold for the features to be extracted.

A positive value of $R$, larger than a predetermined threshold, denotes the presence of a corner while a negative value indicates the presence of an edge and small value indicates a flat region.

The threshold is set depending on the texture of the image.

3.2. Voronoi and Delaunay Tessellations

Voronoi tessellation divides an image into regions where each region consists of all locations in the image whose Euclidean distances are closest to a particular centroid than to any other centroids on the image, in this case the centroids are the feature-points. The boundaries separating two adjacent Voronoi regions containing two centroids is a perpendicular bisector of
the two centroid points. Each vertex of the Voronoi tessellation is the circumcenter the centroids whose regions share the vertex [29].

The Voronoi tessellations are in some instances used to select the regions in which the watermark is embedded [6].

To construct the Delaunay tessellation, nearest neighbouring centroids whose Voronoi regions share an edge are joined. Figure 3.1 shows the Voronoi and Delaunay tessellations of a set of points on a planar surface. The Delaunay tessellations of a set of feature points of images are illustrated in Figure 3.2.

![Fig 3.1](image1.png) **Fig 3.1** (a) The Voronoi tessellation of a set of points and the corresponding (b) Delaunay tessellation.

![Fig 3.2](image2.png) **Fig 3.2** Delaunay triangulations of the feature points for (a) Baboon and (b) Boat test images
The Delaunay tessellation employs the concept of an empty circumcircle criterion which requires the three points forming a Delaunay triangle to lie on the circumcircle and no point should lie inside the circumcircle.

Delaunay tessellation has the property of being unique except in cases of dual circumcircle, which is four points laying on a circumcircle. In such cases then any of the diagonals of the four points can be chosen in order to obtain two triangles [29], [30]. The cells of a planar Delaunay Tessellation are triangular making it simpler to restore than the Voronoi regions which form more complex polygons. The Delaunay tessellations are therefore the obvious choice in the process of restoring a geometrically distorted image back to its original position before attempting to retrieve the embedded watermark in a process called synchronization [25], [26].

The synchronization process involves picking a suitable triangle from the Delaunay triangulation of the original image, called the target triangle, and a corresponding triangle from the attacked image, called the probe triangle. The criteria used to pick the suitable triangles includes checking the angles of the vertices, checking the length of their edges or checking for robustness of the triangles by subjecting them to a variety of distortions. These two triangles are then used to obtain the nature of attack by estimate the angle of rotation, called the Rotation Factor (RF), amount of scaling, called Scaling Factor (SF), and the number of pixels by which the image is translated, called Translation Factor (TF).

### 3.3. Discrete Cosine Transform

The DCT decomposes an image into different frequency bands, allowing for the embedding of information into its mid-frequency sub-bands as shown in Figure 3.3 which shows the most commonly selected coefficients. This is because if information was embedded in the high
Chapter 3: Image Tessellation and DCT/VQ Techniques

frequency bands it would be susceptible to lossy compression and filtering while embedding in the low frequency bands would affect the visual quality of the cover image.

The DCT coefficients $F(u,v)$ of an image function $f(x,y)$ of $M \times N$ pixels are computed as in equation (3.4) [1].

$$F(u,v) = \alpha(u) \alpha(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cos \left( \frac{\pi(2x + 1)u}{2M} \right) \cos \left( \frac{\pi(2y + 1)v}{2N} \right)$$

for $u = 0,1,2,\ldots,M-1$

and $v = 0,1,2,\ldots,N-1$

and the inverse is given by

$$f(x,y) = \alpha(u) \alpha(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(u,v) \cos \left( \frac{\pi(2x + 1)u}{2M} \right) \cos \left( \frac{\pi(2y + 1)v}{2N} \right)$$

for $x = 0,1,2,\ldots,M-1$

and $y = 0,1,2,\ldots,N-1$
where,
\[
\alpha(u) = \begin{cases} 
\frac{1}{\sqrt{M}}, & u = 0 \\
\sqrt{2}, & u \neq 0 
\end{cases} \quad (3.6)
\]

and
\[
\alpha(v) = \begin{cases} 
\frac{1}{\sqrt{N}}, & v = 0 \\
\sqrt{2}, & v \neq 0 
\end{cases} \quad (3.7)
\]

The image is first decomposed into non-overlapping blocks whose sizes are 8 × 8 pixels, as this is the standard of JPEG compression and also gives the best trade-off between computational complexity and image quality degradation. The DCT coefficients are then obtained for each block of pixels. The coefficients are then quantised and normalised by the values, given in Table 3.1, then rounded to the nearest integer. The values in the quantization table are designed to give more resolution to more visually important frequency components than less visually important components. In order to reduce computation complexity, the quantized coefficients are then reordered from the 8×8 blocks into a vector of 64×1 dimension in a zigzag format as shown in Figure 3.4 [1].

### 3.3.1 Embedding Information in DCT Domain

A popular method of encoding secret information in frequency domain is to modulate the relative size of two coefficients in the mid-frequency sub-bands within one image block. One secret message bit is embedded in each 8×8 pixel block. The embedding process starts by selecting a pseudo-random block, this is done by generating a pseudo-random number (PN) using a pseudo-
random number generator (PRNG) which generates random noise with a Gaussian distribution, zero mean ($\mu = 0$) and a variance of one ($\sigma = 1$).

<table>
<thead>
<tr>
<th>(u,v)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td>11</td>
<td>10</td>
<td>16</td>
<td>24</td>
<td>40</td>
<td>51</td>
<td>61</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>12</td>
<td>14</td>
<td>19</td>
<td>26</td>
<td>58</td>
<td>60</td>
<td>55</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>13</td>
<td>16</td>
<td>24</td>
<td>40</td>
<td>57</td>
<td>69</td>
<td>56</td>
</tr>
<tr>
<td>3</td>
<td>14</td>
<td>17</td>
<td>22</td>
<td>29</td>
<td>51</td>
<td>87</td>
<td>80</td>
<td>62</td>
</tr>
<tr>
<td>4</td>
<td>18</td>
<td>22</td>
<td>37</td>
<td>56</td>
<td>68</td>
<td>109</td>
<td>103</td>
<td>77</td>
</tr>
<tr>
<td>5</td>
<td>24</td>
<td>35</td>
<td>55</td>
<td>64</td>
<td>81</td>
<td>104</td>
<td>113</td>
<td>92</td>
</tr>
<tr>
<td>6</td>
<td>49</td>
<td>64</td>
<td>78</td>
<td>87</td>
<td>103</td>
<td>121</td>
<td>120</td>
<td>101</td>
</tr>
<tr>
<td>7</td>
<td>72</td>
<td>92</td>
<td>95</td>
<td>98</td>
<td>112</td>
<td>100</td>
<td>103</td>
<td>99</td>
</tr>
</tbody>
</table>

Table 3.1 Quantisation values of JPEG compression scheme [1].

Two DCT transformed coefficients $F(u_1,v_1)$ and $F(u_2,v_2)$ from the mid-band frequencies are chosen such that their positions correspond to coefficients with equal quantisation values in the JPEG compression scheme given in Table 3.1. This is done to preserve their relative strengths as the two coefficients are scaled by the same value. Some of the candidates for this technique are $F(4,1)$ and $F(3,2)$, $F(1,2)$ and $F(3,0)$ or $F(5,0)$ and $F(2,3)$.

If $F(u_1,v_1) > F(u_2,v_2)$ the block is encoded a “1” otherwise it is encoded a “0”. If the magnitudes of the two selected coefficients do not match with the bit to be encoded in that particular block then the two coefficients are swapped. Since JPEG compression can affect the relative sizes of the coefficients, the algorithm ensures that $|F(u_1,v_1) - F(u_2,v_2)| > k$ for some $k>0$, by adding $k/2$ to
Chapter 3: Image Tessellation and DCT/VQ Techniques

\( F(u_1, v_1) \) and subtracting \( k/2 \) from \( F(u_2, v_2) \). The value of \( k \) gives a trade-off between the robustness of the algorithm and perceptual quality of the image [1].

The encoder then maps the coefficients back to spatial domain by applying inverse DCT. To decode the picture, all available blocks are DCT transformed.

During the watermark extraction, the image is transformed block-wise and the same PN sequence is compared to the values of the transformed block. If the correlation between the sequences exceeds some preset threshold then a “1” is detected.

![Figure 3.4 Zigzag re-ordering of the DCT coefficients.](image)

3.4. Vector Quantization (VQ) Theory

Vector Quantisation has been used for various signal processing applications including compression, recognition and classification [31]. In conventional VQ image compression scheme, a gray level image is decomposed into blocks of size \( m \times n \) with the most commonly used blocks being of sizes \( 4 \times 4, 8 \times 8 \) and \( 16 \times 16 \). Each block is then converted into a vector of
dimension, \( k = mn \). A group of similar vectors are represented by a representative vector called a codeword and the set of all codewords called the codebook [17] and [32]. Basically the codebook is just a lookup table containing codewords and their indices i.e. location in the lookup tables, the index of each codeword is a binary vector. The index code may either have a fixed rate or variable rate. For a fixed rate code of \( b \) bits per input vector, the index has a length of \( b \) bits, however for a variable code rate the indices have variable length and \( b \) is their average length. The compressed image is represented by these indices of fewer bits.

There are three major steps in VQ process namely: codebook design, encoding process and decoding process as shown in Figure 3.5. The most widely used algorithm in codebook design is Linde, Buzo and Gray (LBG) algorithm which uses the principle of the Generalised Lloyd Algorithm (GLA) and was first proposed by Linde et al [32]. It begins by choosing some training vectors from the training set i.e. the block vectors of the input image. The design starts with a small codebook and recursively construct larger ones until the desired size is reached in a process called clustering.

Clustering is broadly classified into classical and fuzzy sets. The classical clustering sets, also called hard or \( k \)-means clustering, are based on crisp decisions in the sense that each input vector can only belong to a single cluster. The fuzzy clustering, also called soft or \( e \)-means clustering, considers each vector as a fuzzy set where a member function is used to determine the possibility of each training set belonging to a cluster. An input may belong to more than one cluster with the degree of certainty being measured by the membership function. This study focuses on the \( k \)-means clustering because of its computational simplicity as compared to the \( e \)-means clustering and it produces satisfactory results in the watermarking process.

After designing the codebook, the VQ encoding process is performed to compress the image. During the encoding process each image vector is replaced by the index of the most appropriate
Chapter 3: Image Tessellation and DCT/VQ Techniques

vector from the codebook. The matching is obtained on the smallest Euclidean distances between the image vector and the codeword of the codebook. The Euclidean distance between two vectors is obtain by equation (3.8). The index of the codebook is therefore transmitted or stored as the compressed image.

\[ d(C, \hat{C}) = \frac{1}{64} \sum_{i=1}^{64} (c_i - \hat{c}_i)^2 \]  

(3.8)

Where, \( C \) is the image vector and \( \hat{C} \) is the codeword.

During decoding each index is used to search for the corresponding codeword in the codebook which is then placed in the position indicated by the index to get decompressed image [32].

In VQ-based watermarking, the watermark can be embedded directly in the codebook of the encoded image and then the extraction is done from the codebook. The watermark can be made

Figure 3.5 Block diagram on VQ encoding and decoding processes
more robust by combining VQ and a transform domain technique. This is done by doing a VQ compression of the image and then obtaining the transform of the codebook, the coefficients of the transformed watermark are then embedded into the coefficients of the transformed codebook [16]. Spread-spectrum can also be used to make the watermark imperceptible to unauthorised users thereby making the embedding process more robust.
CHAPTER 4
GREY-SCALE IMAGE WATERMARKING

4.1. Motivation

Image watermarking has been widely used in copyright protection, however most watermarking techniques are fragile to one type of attack or the other. Geometric distortion attacks are the most notorious as the techniques applied to attain resistance to these attacks are either inadequate or computationally very costly. In an attempt to improve the short comings of the current techniques being employed, the author was motivated to improve on some of the computationally simple techniques while at the same time improving on the robustness of the watermark.

4.2. Image Format

A digital image is a two-dimensional matrix of discrete quantities represented by a function $f(x,y)$, where $x$ and $y$ are spatial coordinate, and the amplitude of $f$ at any pair of coordinates $(x,y)$ is called the intensity of the image at that point. The elements of this matrix are called picture elements, which is often abbreviated as pixels or pels [33].

An image format is a standard way that defines how the image data is organised and stored and describes the arrangement and the type of compression used. Here are a few standards sanctioned by the International Standards Organisation (ISO), the International Electrotechnical Commission (IEC) and the International Telecommunications Union (ITU-T).

i. JPEG

This is a Joint Photographic Experts Group standard for images which is a lossy coding system based on quantized DCT image blocks. It is one of the most popular methods of compressing
images on the internet. A variant of this standard based on quantized DWT compression, called JPEG-2000, has been used in order to increase compression ratio in images.

ii. BMP

This is the Windows Bitmap format that is used mainly for simple uncompressed images.

iii. GIF

The Graphic Interchange Format uses lossless LZW coding for 1 to 8-bit images. It is mostly used in short low resolution films for the World Wide Web.

iv. PNG

The Portable Network Graphics format uses lossless compression to compress full colour images by coding the difference between each pixel’s value and a predicted value based on past pixels.

v. TIFF

The Tagged Image File Format is a flexible file format that supports a variety of image compression standards [33].

4.3. Spatial domain watermarking techniques

In spatial domain watermarking, the insertion of the watermark is done by manipulating the grey-scale levels of an image. The following subsections outline examples of spatial watermarking techniques used by various authors.
Chapter 4: Grey-scale Image Watermarking

4.3.1. Visible watermarking
A popular technique used in embedding a visible watermark was first proposed by Katzenbeisser and Petitcolas [1]. The technique is based on the exploitation of entropy to obtain a visual factor that is used to weigh the watermark for efficient embedding.

Step 1. Divide the original image $I$ and the watermark $W$ into blocks of size 4×4.

Step 2. Compute the visual factor $J$ for each pixel using equation (4.1)

$$J(i, j) = \omega(i, j) \times H(i, j)$$

Where $H(i, j)$ is the entropy used to depict the texture of the image and $\omega(i, j)$ is the contrast sensitivity given by

$$\omega(i, j) = \frac{|I(i, j) - 128|}{128}$$

Step 3. Calculate the scaling factor $\alpha(i, j)$ for each sub-block by using

$$\alpha(i, j) = (b - a) \frac{J(i, j) - \min(J)}{\max(J) - \min(J)} + a$$

Where $a$ and $b$ are predetermined parameters depending on the texture of the image.

Step 4. Select the sub-blocks of the original image for watermark embedding.

Step 5. Modify the pixel value of the selected host image sub-blocks using equation (4.4) to obtain the watermarked image.

$$I'(i, j) = I(i, j) + \alpha(i, j) \times W(i, j)$$

4.3.2. LSB watermarking
The LSB layer of an image has a random distribution and can be used to hide a watermark. A sample procedure proposed by Shoemaker [15] has been outlined below.

Step 1. Normalise the intensity levels of the watermark image in the range [0-1].

Step 2. Decompose the cover image into block of sizes similar to the watermark image size.
Step 3. Embed the watermark into each block using the formula given in equation (4.5) to obtain multiple watermarks into the image.

\[ I^w(x, y) = I(x, y)[1 + \beta w(x, y)] \]  

(4.5)

Where \( I(x,y) \) and \( I^w(x,y) \) are the original and watermarked image coefficients respectively and \( w(x,y) \) is the watermark.

### 4.3.3. CDMA watermarking

The CDMA watermarking technique in the spatial domain is a modification of the LSB substitution method while using a pseudo-random number to determine which bits are substituted. This method of watermarking which improves the robustness of the LSB method as proposed by Katzenbeisser and Petitcolas [1] is outlined below.

**Step 1.** Normalise the intensity levels of the watermark image in the range [0-1].

**Step 2.** Select \( n \) pairs of pixels, corresponding to the number of pixels in the watermark, using a Pseudo-random Number generator where the watermark will be embedded.

**Step 3.** Embed the watermark by slightly increasing or reducing the luminosity contrast of the selected pair as given in equation (4.6).

\[ a'_i = a_i + 1 \]  

\[ b'_i = b_i - 1 \]  

(4.6)

Where \( a'_i \) and \( b'_i \) are the watermarked image pixel pair and \( a_i \) and \( b_i \) are the original image pair.

### 4.4. Transform domain watermarking techniques

This section presents a few methods of transform domain watermarking techniques that have been employed. The various transform domain watermarking techniques presented here are the FMT, DWT and the proposed watermarking scheme which employs VQ-DCT techniques.
4.4.1. **Fourier-Mellin Transform watermarking**

The Fourier-Mellin transform domain is used to embed RST resilient watermarks in images. However, it is not commonly used due to its computational complexity as compared to other geometric distortion focused techniques. The procedure used for image watermarking in the FMT domain as proposed by Kim et al [11] is outlined below.

*Step 1.* Perform the Log-Polar mapping (LPM) of the image.

*Step 2.* Compute the 2D-DFT of the polar coordinates obtained in step 1.

*Step 3.* Embed the watermark into the coefficients using a suitable embedding strength, $\beta$, as described by equation (4.6).

$$I^w(k,v) = I(k,v)[1 + \beta w(x,y)]$$  \hspace{1cm} (4.6)

Where $I(k,v)$ and $I^w(k,v)$ are the original and watermarked image coefficients respectively and $w(x,y)$ is the watermark.

*Step 4.* Perform inverse DFT of the watermarked coefficients and inverse LPM to obtain the watermarked image.

4.4.2. **Discrete Wavelet Transform watermarking**

The watermark can be embedded in the DWT domain in a process described by Shoemaker [15] and outlined below.

*Step 1.* Perform wavelet transform on the image to decompose it into approximation image (LL), horizontal (HL), vertical (LH) and horizontal (HH) detail components.

*Step 2.* Choose a suitable value of $\beta$ to obtain the preferred trade-off between robustness and visual perceptibility.

*Step 3.* Embed the message in the horizontal and vertical detail images as given in equation (4.7).
Chapter 4: Grey-scale Image Watermarking

\[ I^w(u,v) = \begin{cases} 
I(u,v) + \beta w(x,y), & u,v \in HL,LH \\
I(u,v), & u,v \notin LL,HH
\end{cases} \]  \hspace{2cm} (4.7)

Where, \( I(u,v) \) and \( I^w(u,v) \) are the original and watermarked DWT coefficients of the image and \( w(x,y) \) are the pixels of the watermark.

**Step 4.** Perform inverse DWT to obtain the watermarked image.

### 4.4.3. Proposed watermarking technique

The watermark embedding process employed in the research for this thesis involved embedding a 25 × 40 pixel binary image into 256 × 256 and 512 × 512 pixel grey-scale images with pixel resolutions of 8-bits. The embedding procedure is described as follows [29].

**Step 1.** The LBG algorithm is used to construct a codebook of the image with a codeword size of 8×8.

**Step 2.** Decompose the image into 8×8 blocks of the codebook and compute a block-based DCT of the codebook and select the coefficients of the mid-frequency sub-band.

**Step 3.** Generate a pseudo-random sequence and select the value of \( \beta \) which controls the embedding strength and gives a trade-off between robustness and perceptual quality.

**Step 4.** Embed the watermark into the selected mid-frequency sub-band using the pseudo-random (PN) sequence to determine the coefficients where the watermark is embedded as follows:

\[ I^w(u,v) = \begin{cases} 
I(u,v)[1 + \beta W(u,v)], & u,v \in F_M \\
I(u,v), & u,v \notin F_M
\end{cases} \]  \hspace{2cm} (4.8)

Where, \( I(u,v) \), \( I^w(u,v) \) and \( W(u,v) \) are the original, watermarked and watermark DCT coefficients of the codebook respectively and \( F_M \) is the mid-frequency sub-band.

**Step 5.** Compute the inverse DCT of the watermarked codebook and then perform VQ decoding to obtain the watermarked image.
Chapter 4: Grey-scale Image Watermarking

The value of $\beta$ between 0.1 and 0.5 was found to give a fair trade-off between robustness and visual quality depending on the pixel texture of the image [30]. It was observed that the generally smooth pixel images a higher value of $\beta$ could be used with less visual quality distortion than the rough texture images. The full MATLAB code is given in Appendix A.

4.5. Watermark Extraction

The watermark extraction procedure used in this thesis research is outlined as follows [28].

*Step 1.* The feature-points of the image are extracted using the Harris corner detector and then the Delaunay tessellation of the feature-points is obtained.

*Step 2.* Several target triangles and corresponding probe triangles are obtained from the Delaunay tessellations of the original and attacked image respectively. The triangles are by selecting the robust triangles that have long edges.

*Step 3.* Using the triangles, the amount of distortion that each probe triangle has undergone is estimated by comparing the orientation angles, sizes and positions to the corresponding target triangle.

*Step 4.* The average of the distortions undergone by all the triangles is obtained and used as the rotation factor ($RF$), scaling factor ($SF$) and translation factor ($TF$) which are then used to restore the image to its original position.

*Step 5.* The codebook of the resynchronized image obtained using the same algorithm used during the embedding process and its DCT is computed.

*Step 6.* The mid-band frequencies are selected and using the same PN sequence used in the embedding process the coefficients are checked for the presence or absence of the embedded watermark by correlating the coefficients of the mid-band frequencies and the PN sequence.

*Step 7.* The watermark is finally reconstructed using the extracted watermark bits.
Chapter 4: Grey-scale Image Watermarking

The MATLAB code for watermark extraction is given in Appendix A.

4.6. **Note on Publication**

A paper entitled “A Robust Image Watermarking Scheme Invariant to Rotation, Scaling and Translation Attacks,” [28] was presented at the 16th IEEE Mediterranean Electrotechnical Conference (MELECON 2012) which took place from 25th to 28th of March 2012 in Yasmine – Hammamet, Tunisia. The paper has been published in IEEE Xplore under the proceedings of the 16th IEEE Mediterranean Electrotechnical Conference (MELECON 2012) from page 379 to 382. The published paper is given in Appendix B.
Any of the grey-scale image watermarking techniques can also be applied in watermarking coloured images. However, watermarking coloured images requires pre-processing as they are composed of various components depending on the colour space. The watermark is embedded in different components of the colour spaces. A brief presentation of the basics of chromaticity and colour spaces is given in this chapter and appropriate watermarking techniques discussed. In addition, a presentation of video formats is given as well as the watermarking techniques commonly used.

5.1. **Motivation**

In recent years, the availability and use of colour images has surpassed that of grey-scale images. The author was motivated to test if the proposed technique would attain the robustness and perceptual fidelity it exhibited in grey-scale images if it were to be applied to colour images. Simulation tests were then performed to check if there was an improvement in robustness using the proposed technique as compared to other related techniques. The author was then motivated to try video watermarking and attempt to improve on existing techniques.

5.2. **Basics of Chromaticity**

Visible light can be categorised into achromatic and chromatic light. Achromatic light only contains information of intensity and no information on the colour of the image presented and is also referred to as grayscale. The electromagnetic spectrum of wavelengths in the range of 400 to
700nm consists of chromatic light, where blue has the shortest wavelength of 435.8nm, green has a wavelength of 546.1nm and red a wavelength of 700nm [33].

Colours are generally distinguished by their brightness, hue and saturation, where brightness is the information on intensity, hue is an attribute of the dominant colour and saturation is the relative purity of the dominant colour. Since hue and saturation are the properties that attribute the colour information of light, they are collectively referred to as chromaticity.

A colour can be specified using the chromaticity diagram in Figure 5.1 where its position in the diagram can be used to specify its hue and saturation. The x-axis represents red while y-axis represents green. The colours that lie on the boundary represent the full range of the chromatic light spectrum and are in their purest form and the saturation is decreased as the colour approaches the white-spot marked by D65 in Figure 5.2. The colours found within the triangle give the full range of colours that can be obtained from mixing the three primary colours additively. The white-spot represented by D65 lies on 6504K spot on the Planckian locus represented by $T_c(K)$ which represents the colour a black body would take as its temperature changes also known as correlated colour temperature. The correlated colour temperatures related to direct daylight range from 4000K represented as D40 to 7000K represented as D70 [3].
Chapter 5: Colour Image and Video Watermarking

Figure 5.1 Chromaticity diagram (Courtesy of CIE)

Figure 5.2 Typical range of colours produced by colour monitors
5.3. **Colour Spaces**

5.3.1. **RGB colour space**

The RGB colour images are $M \times N \times 3$ arrays of colour pixels, where each pixel is a triplet consisting of red, green and blue components at specific spatial locations. It can be viewed as a stack of three gray-scale images passed through red, green and blue screens. The images of an RGB colour are called the red, green and blue *component images*. This colour space is based on the Cartesian coordinate system and is represented by the RGB cube given in Figure 5.3. The three component colours are at three corners of the cube and are known as primary colours of light since when mixed in various intensity proportions and reflected can produce any visible colour. The complement of the three primary colours are found at the other three corners of the cube and are known as secondary colours. Black is found at the origin and white at the corner furthest from origin and grayscale is obtained from the line that connects these two corners.

All the components of an RGB image contain information on both the intensity and colour of the image [33]. Figure 5.4 shows an image in the RGB colour space and in the CMY space.

*Figure 5.3 The RGB cube*
5.3.2. CMY and CMYK colour space

This colour space is composed of Cyan, Magenta and Yellow components. In some cases especially in printing applications a fourth ink that is black is added and is denoted by the letter ‘K’ which stands for Key. This colour space is a complement of the RGB colour space or in other words may be viewed that if red, green and blue are primary colours of light then cyan, magenta and yellow are secondary colours of light. They are also known as the primary colours of pigment since they absorb their corresponding primary colours of light. That is if white light is incident on a yellow pigment, its blue component is absorbed and not reflected back.

Being complements of RGB, when the cyan, magenta and yellow colours are added in equal proportions, they should produce a black colour which is the complement of the addition of RGB in equal proportions. In printers however, this mixture results into a brownish colour hence a separate black ink cartridge is usually included hence the CMYK colour space.

Conversion of RGB to CMY components is done using the transform given in equation (5.1) assuming the colours are normalised to the range of [0,1].
\[
\begin{bmatrix}
C \\
M \\
Y
\end{bmatrix} =
\begin{bmatrix}
1 \\
1 \\
1
\end{bmatrix} -
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix}
\]  

(5.1)

5.3.3. National Television System Committee (NTSC) colour space

The National Television System Committee (NTSC) colour space is mostly used in the United States of America for television broadcasting. It has the advantage of gray-scale information being separate from the colour information hence the same signal can be used for both colour and monochrome television sets. It also consists of three components: luminance \((Y)\), hue \((I)\) and saturation \((Q)\) components hence it is at times referred to as the \(YIQ\) colour space. The luminance component, also called the \textit{Luma}, contains the gray-scale information of the image while the colour information is contained in the hue and saturation components, the \textit{Chroma}.

The NTSC colour space can be obtained by converting the RGB component to YIQ components using the transformation given in equation (5.2) \([33]\) with the components normalised in the range \([0,1]\).

\[
\begin{bmatrix}
Y \\
I \\
Q
\end{bmatrix} =
\begin{bmatrix}
0.299 & 0.587 & 0.114 \\
0.596 & -0.274 & -0.322 \\
0.211 & -0.523 & 0.312
\end{bmatrix}
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix}
\]  

(5.2)

It can be seen from the equation (5.2) that the intensity information, \(Y\), is obtained by adding the intensities of the RGB components, this can be seen from the addition of the matrix elements in the first row. Since grayscale images have a balance in the RGB components the elements of the matrix used in determining colour add up to zero. The RGB components can also be obtained from the YIQ components using the equation (5.3).

\[
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix} =
\begin{bmatrix}
1 & 0.956 & 0.621 \\
1 & -0.273 & -0.647 \\
1 & -1.104 & 1.701
\end{bmatrix}
\begin{bmatrix}
Y \\
I \\
Q
\end{bmatrix}
\]  

(5.3)

The luminance component is reported to give the best results when used for watermark embedding as compared to the hue and saturation components of an NTSC colour image \([34]\).
Figure 5.5 shows the Peppers image in RGB colour space and when it is converted to the NTSC colour space.

![Peppers image](image1)

![Peppers image (b)](image2)

*Figure 5.5 Peppers image (a) in RGB colour space and converted to (b) NTSC.*

### 5.3.4. YCbCr colour space

YCbCr colour space is composed of luma ($Y$) and chroma components ($Cr$ and $Cb$) which give the gray-scale and colour information of the image respectively. The chroma components, $Cr$ and $Cb$, are subtractive colour of red and blue respectively. This colour space is commonly used in digital video files [33].

The YCbCr components can be obtained from the RGB colour space by using the transformation given in equation (5.4).

\[
\begin{bmatrix}
Y \\
Cb \\
Cr
\end{bmatrix} = \begin{bmatrix}
16 \\
128 \\
128
\end{bmatrix} + \begin{bmatrix}
65.481 & 128.553 & 24.966 \\
-37.797 & -74.203 & 122.000 \\
112.000 & -93.786 & -18.214
\end{bmatrix} \begin{bmatrix}
R \\
G \\
B
\end{bmatrix}
\]  

(5.4)

The best results of watermarking are obtained when the luminance component $Y$ is employed for watermarking [35]. Figure 5.6 shows the RGB image and its conversion into YCbCr colour space.
5.3.5. HSV colour space
The HSV colour image also consists of three components: Hue (H), Saturation (S) and Value (V) components. It is formulated by looking at the RGB cube along its gray (diagonal) axis. This colour scheme closely resembles human perception of colour images as the terms hue, saturation and value can be approximated as tint, shade and tone. The Value component gives the best results when used for embedding the watermark [34]. The HSV colour space can be obtained from the RGB colour system using the expressions given in equations (5.5) to (5.7) [33].

\[ H = \begin{cases} \theta^\circ, & B \leq G \\ 360^\circ - \theta^\circ, & B > G \end{cases} \]  

(5.5)

Where \( \theta \) is given by

\[ \theta = \cos^{-1} \left( \frac{\frac{3}{2}[(R - G) + (R - B)]}{\left[ (R - G)^2 + (R - B)(G - B) \right]^\frac{1}{2}} \right) \]  

\[ S = 1 - \frac{3}{(R+G+B)}[\min(R,G,B)] \]  

(5.6)

\[ V = \frac{1}{3} (R + G + B) \]  

(5.7)
Chapter 5: Colour Image and Video Watermarking

It is assumed in the expressions that RGB values are normalised to the range of [0,1] and \( \theta \) is measured with respect to the red axis.

The HSV colour system can be converted to RGB colour space using the expressions given in equations (5.8) to (5.10).

\[
B = V (1 - S) \quad (5.8)
\]

\[
R = V \left[ 1 + \frac{S \cos H}{\cos (60^\circ + H)} \right] \quad (5.9)
\]

\[
G = 3V - (R + B) \quad (5.10)
\]

However, since Hue is in the interval of \([0^\circ, 360^\circ]\) it is divided into three intervals representing each of the three sectors known as the RG sector \((0^\circ \leq H < 120^\circ)\), the GB sector \((120^\circ \leq H < 240^\circ)\) and the BR sector \((240^\circ \leq H \leq 360^\circ)\). The value of \(H\) used for each of these sectors has a range of 120° and is calculated as follows: For the RG sector the actual value of \(H\) is used, while \(H = H - 120^\circ\) is used in the GB sector and \(H = H - 240^\circ\) is used in the BR sector [33]. Figure 5.7 shows the RGB image and when it is converted to HSV colour space.

![Peppers image in (a) the RGB colour space and converted to (b) HSV.](image)

*Figure 5.7 Peppers image in (a) the RGB colour space and converted to (b) HSV.*
5.3.6. HSI colour space

The HSI colour space is composed of *Hue*, *Saturation* and *Intensity* components. It is closely related to the HSV colour space, the difference being that the HSI colour space gives the intensity in the form of an artist’s perception while HSV assigns the intensity a numerical value. Similar to the HSV colour space, the best component to hide the watermark information is the intensity component [34].

5.4. Video Formats

Video formats are categorised by the standard of compression used in the coding process and the organisation that has sanctioned the standard. Just like the image compression standards the sanctioning organisations for video coding standards are ISO, IEC and ITU-T. The subsections below give a brief overview of the video coding standards as summarised by Gonzalez and Woods [33].

i. **H.261**

This is an ITU-T’s coding standard developed for two-way videoconferencing for Integrated Services Digital Network (ISDN) lines. It supports non-interlaced $352 \times 288$ and $176 \times 144$ pixel images, known as Common Intermediate Format (CIF) and Quarter CIF (QCIF) respectively. A compression approach similar to JPEG is used to reduce spatial redundancy while frame-to-frame prediction is used to reduce temporal redundancy and block-based motion compensation technique is used.

ii. **MPEG-1**

This is the ISO/IEC’s Motion Pictures Experts Group’s standard for CD-ROM applications with non-interlaced video. It is similar to H.261 but frame predictions are based on previous frame, next frame, or an interpolation of both.
iii. H.262/MPEG-2

This is an extension of MPEG-1 designed for DVDs and optimized for TV broadcast with higher transfer rates. This standard supports interlaced videos whose image quality is similar to NTSC, PAL and SECAM of resolution 704×480 Pixels at 4 - 8 Mbit/s and HDTV at 20 Mbit/s making it the most widely used compression engine for internet video streaming. MPEG-2 coding standard includes block wise motion compensation of frames/pictures with blocks of size 16×8 pixels. The coding efficiency is further improved by using different quantization and Variable Length Coding (VLC) tables and various scalability modes. Use of different quantization and VLC tables for each block enables the coding process to use fewer bits on less visually important blocks as compared to the more visually important blocks of each frame/picture.

iv. H.263

This is an enhanced version of the H.261 designed for ordinary telephone modems with additional resolutions to include Sub-QCIF at frame rates usually below 10 fps. H.263 is widely used as a compression engine for Internet video streaming and is the compression core of the MPEG-4 standard.

v. H.264/MPEG-4 part 10

An extension of H.261-H.263 and MPEG-2 designed for video conferencing, Internet streaming and television broadcasting. It supports prediction differences within frames, integer transform and context adaptive arithmetic coding [36].

5.5. Feature-Points Detection and Tessellation in Colour Images

The process of extracting feature-points and Delaunay tessellation in colour images is similar to the processes used in grey-scale images regardless of the colour space and the processes are illustrated in sections 3.1 and 3.2 of chapter 3.
5.6. VQ and DCT in Colour Images

Colour images can be quantised directly in a process known as Colour Quantisation or decomposed into its component images before applying VQ in each component in the same way as grey-scale images. Colour Quantisation uses the same principle as VQ only that the codewords in this case are used to represent specific colours. Due to the computational complexity of Colour Quantisation, VQ is used in this study to compress colour images. The HSV colour space is chosen to illustrate the VQ and DCT process applied on colour images during the course of this study. HSV colour space is chosen to represent all the colour spaces which have separate Luma and Chroma components where the luma components gave the best component for embedding the watermark as seen in section 6.4.

5.6.1. VQ in Colour Images

The VQ process has been reported to improve the robustness of the watermarking process [16], [17], [29], and was therefore applied in watermarking. To limit the complexity of the algorithm, the VQ process was performed only to the component that was chosen for embedding the watermark. Since HSV colour space was chosen to illustrate the VQ process in this report, where it is most suitable to embed the watermark in the Value component. The process is performed as outlined below.

*Step 1.* Extract the Hue, Saturation and Value components from the colour image.

*Step 2.* Decompose the Value component into blocks of size $8 \times 8$ then convert the blocks into vectors of size $64 \times 1$, $C_i$, $i=1,2,\ldots,4096$.

*Step 3.* Set the representative vectors (codewords) depending on the desired codebook size, $N$. $\hat{C}_i$, $i=1,2,\ldots,1024$. 
Step 4. Each vector is compared to the representative vectors and the best matching vector selected using distortion rule as given below.

Choose $C_k$ such that $d(C, \hat{C}_k) \leq d(C, \hat{C}_j)$ for all $j=1, \ldots, 1024$.

Where $d(C, \hat{C})$ denotes the square of the Euclidean distance between original vector $C$ and representative vector $\hat{C}$ given by

$$d(C, \hat{C}) = \frac{1}{64} \sum_{i=1}^{64} (c_i - \hat{c}_i)^2$$

(5.11)

Where $c_i$ and $\hat{c}_i$ are the $i^{th}$ elements of original vector $C$ and representative vector $\hat{C}$ respectively.

Step 5. Calculate the centroid of each region formed from the above process and use the centroid as the new representative vector. Repeat steps 4 and 5 until there is minimum distance between the obtained centroid and the representative vector.

Step 6. Terminate and used the centroids obtained as the codebook.

5.6.2. DCT in colour images

During the research reported in this thesis, watermark is embedded in the DCT domain into the codebook of the Value component of the colour image. Just like in the grey-scale images, the codebook is decomposed into $8 \times 8$ blocks before performing a block based DCT as described in section 3.3.

5.7. Watermark Embedding in Colour Images

The process of watermark embedding in colour images are performed as described below.

Step 1. Extract the colour components and select the suitable component for the embedding process, e.g. Value component of HSV colour space.
Step 2. The LBG algorithm is used to construct a codebook of the component image with a codeword size of 8×8.

Step 3. Decompose the image into 8×8 blocks of the codebook and compute a block-based DCT of the codebook and select the coefficients of the mid-frequency sub-band.

Step 4. Generate a pseudo-random sequence and select the value of $\beta$ which controls the embedding strength and gives a trade-off between robustness and perceptual quality.

Step 5. Embed the watermark into the selected mid-frequency sub-band using the pseudo-random (PN) sequence to determine the coefficients where the watermark is embedded (5.12)

\[ I^w(u, v) = I(u, v) + \beta W(u, v) \]

Where, $I(u,v)$, $I^w(u,v)$ and $W(u,v)$ are the original, watermarked and watermark DCT coefficients of the codebook respectively.

Step 6. Compute the inverse DCT of the watermarked codebook and then perform VQ decoding to obtain the watermarked image component which is concatenated to the other image components to obtain a watermarked colour image.

5.8. Watermark Extraction in Colour Images

The extraction of the watermark is then performed through the following process.

Step 1. The feature-points of the colour image are extracted using the Harris corner detector and then the Delaunay tessellation of the feature-points is obtained.

Step 2. Several target triangles and corresponding probe triangles are obtained from the Delaunay tessellations of the original and attacked colour images respectively.
Chapter 5: Colour Image and Video Watermarking

Step 3. Using the triangles, the amount of distortion that each probe triangle has undergone is estimated by comparing the orientation angles, sizes and positions to the corresponding target triangle.

Step 4. The average of the distortions undergone by all the triangles is obtained and used as the rotation factor (RF), scaling factor (SF) and translation factor (TF) which are then used to restore the image to its original position.

Step 5. The restored colour image is then decomposed to its components and the component used for embedding is selected for further processing.

Step 6. The codebook of the selected component image is obtained using the same algorithm used during the embedding process and its DCT is computed.

Step 7. The mid-band frequencies are selected and using the same PN sequence used in the embedding process the coefficients are checked for the presence or absence of the embedded watermark by correlating the coefficients of the mid-band frequencies and the PN sequence.

Step 8. The watermark is finally reconstructed using the extracted watermark bits.

5.9. Watermark Embedding and Extraction in Videos

Just like still images, video files need copy as well as copyright protection and finger-printing. Video files, however, require more preprocessing than the image files as they consist of video data, audio data and time component.

While watermarking a video file, first the video data is separated from the audio data by a demultiplexer. The audio data could also be watermarked however it is not investigated further in this study as the study’s main focus lies in image watermarking. The video data, which consists of images called pictures. A picture consists of one or two frames depending on the type of scanning applied in the projection of the video. If the video uses a progressive scan then a
picture consists of one frame while interlaced pictures have two frames; one for the odd lines and the other for even lines. The frames are further preprocessed before the embedding process commences [37].

The video data consists of three types of frames. These are the intra-coded frames also known as I-frames, the inter-coded frames which consist of bi-predictive (B) frames, and the predictive (P) frames. The I-frames do not depend on any reference pictures for prediction therefore contain more information than the inter-coded frames which depend on other frames for reference picture.

The I-frames are the most suitable for watermarking as they are the least compressed hence less likely to suffer compression attacks. These frames are also most the visually significant frames in a video file hence if attacked may compromise the visual quality of the video file [38].

The watermark embedding process is then carried out in the selected frames in the same manner in which it is done in the colour images as outlined in the previous section. The frames are then concatenated to form the video data which is multiplexed with the audio data resulting into a watermarked video file [38], [39].

However video watermarking poses additional challenges that are not encountered in image watermarking. In the most recent coding standard, ITU-T’s H.264/AVC or ISO’s MPEG part 10, the watermark embedding process is complicated by the fact that only visually significant information is left in the frames after the coding process. This makes it hard to embed any information without degrading the visual quality of the video. This is further complicated by the fact that prediction coding is done at slice level rather than frame level as was in the previous coding standards [40]. Also in addition to the signal processing and geometric distortion attacks encountered in image watermarking, video watermarks encounter spatial and temporal collision attacks and frame drop attacks. The temporal attacks correlate the frames to check for similar
data in uncorrelated frames or uncorrelated data in subsequent and highly correlated frames. In other words, if subsequent frames of a scene have different watermarks or if different scenes have the same watermark then temporal collision can be used to detect the positions of the watermark [41].

Research is being carried out on the use of motion prediction and motion compensation in the watermarking process to address most of the attacks unique to video watermarking. Also the use of a combination of the above and embedding in relatively high frequency regions of the frame images such as edges and corners to reduce the impact of watermarking on the quality of the image [42].

Other techniques proposed to counter the problem of collision attacks include the use of complementary watermark messages so that if the attacker finds one message they may miss the other and embedding watermark in transitional frames which do not correlate to any other frames [38].

5.10. Note on Publication

The work on colour image watermarking was accepted and presented at the IEEE Africon 2013 Conference held on 9th to 12th September 2013 in Mauritius. The paper entitled “A Colour Image Watermarking Technique Resistant to Affine Geometric Attacks” [43] has been indexed and published in the IEEE Xplore digital library under the proceedings of the IEEE Africon 2013 conference and is also annexed in Appendix C.
A variety of common test images were used in simulating and testing the various techniques in this thesis research. The images were used to test the perceptibility and robustness of the proposed technique on images of varying textures, sizes and phenomena in the images. The proposed technique is then compared to other related techniques proposed by other authors. The cover images used were of sizes 256 × 256 and 512 × 512 pixels which had resolutions of 8-bits and 24-bits per pixel for grey-scale and colour images respectively. They included common test images such as Cameraman, Lena, Baboon, Barbara, Peppers, lifting body and Boat among others. Several videos were also used to simulate the video watermarking process and they included: Container, Foreman, Bus, Carphone and an extract from the movie Cheaper by the Dozen. The videos used in the simulation had image resolutions of 176 × 144, 352 × 288 and 704×576. The watermark image used is a 25 × 40 pixel binary logo obtained from the internet [44].

MATLAB was used in the simulation of the watermarking scheme and its performance under different kinds of attack and the H.264/14496-10 AVC JM reference software version 16.0 was used to code the video clips during the simulation process [45].

6.1. Quality Measures

There are numerous metrics used in determining the level of image degradation after watermark has been embedded. The metrics include the Mean Squared Error and Peak Signal-to-Noise Ratio.
Although it does not give an account of the perceptible degradation to the cover-image, Peak Signal-to-Noise Ratio (PSNR) is the most widely used metric in finding the quality of a watermarked image [10] and is defined as given in equation (6.1).

\[
PSNR_{dB} = 20 \log_{10} \left( \frac{MAX_I}{MSE} \right) \text{dB}
\]  

(6.1)

where, \( MAX_I \) is the highest grayscale level of a pixel and \( MSE \) is the Mean Square Error which is given by [10]

\[
MSE = \sum_{i=1}^{N} (X_i - X'_i)^2
\]  

(6.2)

where \( X_i \) and \( X'_i \) are the original and the watermarked images respectively.

The quality of the extracted watermark can be computed using Normalized Cross-correlation (NC), this can be used in the evaluation of the effectiveness of the algorithm. The NC between the embedded watermark, \( W(i,j) \), and the extracted watermark, \( \hat{W}(i,j) \), is given by equation (6.3).

\[
NC = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} [W(i,j) \cdot \hat{W}(i,j)]}{\sum_{i=1}^{M} \sum_{j=1}^{N} [W(i,j)]^2}
\]  

(6.3)

where \( M \times N \) is the size of the cover image.

The value of NC lies between 0 and 1, the larger the value the better the watermark extracted.

\textbf{6.2. Computer Simulation Results}

The results give the outcome of experiments performed to evaluate the performance of the proposed technique by determining the perceptibility and robustness of the embedded watermark in comparison to other related techniques. The \( PSNR \) values are used to determine the perceptibility of the watermark by assessing degradation in the quality of the watermarked image and \( NC \) factor are used to determine the robustness of the embedded watermark. The following subsections give a presentation of the computer simulation results.
6.3. **Grey-scale image watermarking results**

Computer simulations experiments were performed on several grey-scale images to test the perceptibility and robustness of the watermarking process. The perceptibility was tested by obtaining the PSNR of the watermarked image while robustness was tested by simulating a variety of attacks on the watermarked image and obtaining the NC factor of the extracted watermark.

6.3.1. **Visual perceptibility test**

Figure 6.1 shows the cover and watermarked images of *Cameraman* of 256 × 256 pixels and 256 intensity levels while Figure 6.2 shows the cover and watermarked images of *Lena* which have a resolution of 512 × 512 pixels and 256 intensity levels. The images in Figures 6.1 and 6.2 are the results obtained while testing for the visual perceptibility of the watermark. The two images were chosen in order to demonstrate the results obtained when the technique is used on images that have large regions of low frequency and man-made features, such as in *Cameraman*, as compared to images dominated with natural features and have high frequency components respectively which are found in *Lena*.

Since the embedding processes resulted into watermarked images with little or no visual degradation, the perceptibility of the watermark was deemed satisfactory and it was then tested for robustness. This is evidence by the fact that the two images i.e. Figure 6.1 (a) and (b) are indistinguishable, so are the pairs Figure 6.2 (a) and (b).
Chapter 6: Results and Discussions

Figure 6.1 Cover image (a) and watermarked image (b) at a PSNR of 34.9 dB

Figure 6.2 Cover image (a) and Watermarked image (b) at a PSNR of 38.7 dB

6.3.2. Robustness tests

To test for robustness, a variety of signal processing and geometric distortion attacks were simulated on the watermarked images and the robustness of the extracted watermark evaluated by computing the NC factor.
Chapter 6: Results and Discussions

i. JPEG compression attack

Figures 6.3, 6.4 and 6.5 show the images of Cameraman and Lena subjected to a JPEG compression attack at a quality factor 50% and 30% respectively, the embedded message and the extracted message.

JPEG compression is a lossy compression which is done by zonal masking and scalar quantization. JPEG compression first begins with performing a block-wise DCT of the image, then each coefficient is multiplied by the corresponding element in a zonal mask. The zonal mask is a matrix consisting of 1’s in locations where maximum variance is required and 0’s in the remaining locations. Then a quantisation of the result of zonal masking is done using the JPEG quantisation table. The JPEG quantisation results into removal of some frequency components of an image [33].

The quality of the cover images was reduced to unacceptable levels while the extracted watermarks had insignificant degradation. It can therefore be concluded that the technique proposed in this thesis is robust to JPEG compression attacks.

Figure 6.3 JPEG Compression attacked Cameraman image (a) with a Quality Factor of 50%, the embedded watermark (b) and the retrieved watermark (c) with NC of 0.99.
Chapter 6: Results and Discussions

Figure 6.4 JPEG Compression attacked Lena image (a) with a Quality Factor of 50% and the retrieved watermark (b) with NC of 0.99.

Figure 6.5 JPEG Compression attacked images with a Quality Factor of 30%, and the respective retrieved watermarks with NC of 0.98 and 0.99.
ii. **Histogram equalisation attack**

Figure 6.6 illustrates the watermarked images after they had been subjected to a histogram equalisation attack and the extracted watermark. It can be seen in the figure that even though the cover images are significantly degraded, the quality of the extracted watermarks are still acceptable.

Histogram equalization is a technique for adjusting image intensities to enhance contrast. In order to perform histogram equalisation on an image, the probability mass function of all the pixels in the image are first calculated. Then the cumulative distribution function is calculated according to gray levels. The CDF value is then used to map the new gray levels into a number of pixels in order to equalize the histogram of the image.

![Cameraman Image](image1.png)  ![Lena Image](image2.png)

*Figure 6.6 Histogram Equalisation attacked Cameraman (a) and Lena (b) images and the extracted watermarks respectively of NC 0.97.*
The proposed watermarking scheme is thus seen to be robust to histogram equalisation attack as evidenced in the result given in Figure 6.6. Simulations were also performed to test the robustness of the proposed technique to average filtering and unsharp filtering and the results presented in subsection iv of this section.

**iii. RST attack tests**

Since the DCT embedding algorithm exhibited sufficient robustness to signal processing attacks, it was tested for its robustness to geometric distortion attacks. Figure 6.7 illustrates the extracted watermark when the watermarked image has been subjected to rotation, scaling and translation attacks and the watermark recovery process done without attempting to reverse the attack on the watermarked image as proposed by Pal et al [17].

In order to test for the effect of synchronising the attacked image Delaunay triangulation was used in the synchronisation process. Figure 6.8 illustrates the Delaunay triangulation used in synchronising attacked images to their original positions.

![Figure 6.7 Extracted watermarks from images subjected to Geometric Distortion attacks.](image)

(a) (b) (c) (d) (e) (f) (g)

Figures 6.9 and 6.10 show watermarked images subjected to rotation attacks and the angle of rotation estimated using the method proposed by Tang and Hang [26] while Figures 6.11 and
Chapter 6: Results and Discussions

6.12 show the technique proposed in this thesis and the message extracted in each case. The attacked image was rotated at an RF of $39.4^\circ$ in the simulation process.

![Figure 6.8 Delaunay triangulations used in the synchronisation process](image)

Figure 6.8 Delaunay triangulations used in the synchronisation process

![Figure 6.9 (a) The Delaunay triangulation of Cameraman image, (b) the probe triangle used in estimating angle of rotation as proposed by Tang [26] and (c) the extracted watermark at NC of 0.93. Angle of rotation (RF) estimated at 39°.](image)

Figure 6.9 (a) The Delaunay triangulation of Cameraman image, (b) the probe triangle used in estimating angle of rotation as proposed by Tang [26] and (c) the extracted watermark at NC of 0.93. Angle of rotation (RF) estimated at 39°.
Chapter 6: Results and Discussions

Figure 6.10 (a) The Delaunay triangulation of Lena image, (b) the probe triangle used in estimating angle of rotation of Lena image as proposed by Tang [26] and (c) the extracted watermark at NC of 0.90. Angle of rotation (RF) estimated at 65°.

Figure 6.11 Probe triangles used in (b) the proposed technique to estimate the angle of rotation estimated at 39.5° and (c) the extracted watermark with NC of 0.99.
Chapter 6: Results and Discussions

Figure 6.12 (a) Delaunay triangulation, (b) the probe triangles used in the proposed technique to estimate the angle of rotation estimated at 64.6° and (c) the extracted watermark with NC of 0.99.

The robustness of the proposed scheme was tested by subjecting the test images to random angles of rotation and comparing the quality of the extracted watermark with various related techniques. The comparison the proposed technique and the closely related techniques to rotation attacks is given by the plots in Figures 6.13 and 6.14. It is evident from the plots that the proposed technique performed much better as the angles of rotation approached odd multiples of 45°. The NC values of the proposed technique remained above 0.95 where as they fell lower for the other techniques when the angles of rotation produced dark regions around the image.
Chapter 6: Results and Discussions

Figure 6.13 Plot showing comparison of quality of extracted message between Tang’s, Pal’s and the proposed method after the watermarked Cameraman image subjected to various angles of rotation attack.

Figure 6.14 Plot showing comparison of quality of extracted message between Tang’s, Pal’s and the proposed method after the watermarked Lena image subjected to various angles of rotation attack.

Key: Prop. = proposed technique, Tang = Tang and Hang’s technique and Pal = Pal et al technique.
iv. Summary of robustness of the proposed technique

Cropping, scaling and translation attacks were also simulated by extracting the embedded watermark using Tang’s technique, Pal’s technique and the proposed technique and a comparison of the techniques is summarised in Table 6.1. A graphical comparison of the quality of the watermark extracted from the Baboon image using various techniques after it been subjected to attack is given in Figure 6.15.

It is evident in the table that the three techniques performed equally well when signal processing attacks were simulated on the cover images. This is attributed to the use of the DCT as the embedding transform for all the techniques and its robustness to signal processing attacks [10], [14], and [15]. However, the techniques varied in their performance when it came to geometric distortion attacks. When the cover image is exposed to cropping attack, the proposed technique and Pal’s technique exhibited a higher robustness as compared to Tang’s technique.
Table 6.1 A comparison of the performance of the proposed watermarking technique with some related techniques.

<table>
<thead>
<tr>
<th>Attack method</th>
<th>Cameraman</th>
<th>Lena</th>
<th>Baboon</th>
<th>Peppers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prop</td>
<td>Tang</td>
<td>Pal</td>
<td>Prop</td>
</tr>
<tr>
<td><strong>JPEG (QF = 30%)</strong></td>
<td>0.98</td>
<td>0.96</td>
<td>0.97</td>
<td>0.99</td>
</tr>
<tr>
<td><strong>JPEG (QF = 50%)</strong></td>
<td>0.99</td>
<td>0.98</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td><strong>Histogram Equal.</strong></td>
<td>0.97</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
</tr>
<tr>
<td><strong>Average Filtering</strong></td>
<td>0.97</td>
<td>0.97</td>
<td>0.98</td>
<td>0.98</td>
</tr>
<tr>
<td><strong>Unsharp Filtering</strong></td>
<td>0.99</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
</tr>
<tr>
<td><strong>Cropping 0.1</strong></td>
<td>0.89</td>
<td>0.72</td>
<td>0.87</td>
<td>0.93</td>
</tr>
<tr>
<td><strong>Rotation (39.4°)</strong></td>
<td>0.99</td>
<td>0.93</td>
<td>0.76</td>
<td>0.99</td>
</tr>
<tr>
<td><strong>Scaling (0.5)</strong></td>
<td>0.94</td>
<td>0.87</td>
<td>0.67</td>
<td>0.91</td>
</tr>
<tr>
<td><strong>Translation (10,0)</strong></td>
<td>0.98</td>
<td>0.97</td>
<td>0.79</td>
<td>0.97</td>
</tr>
<tr>
<td><strong>RST 25°, 0.5, (10,5)</strong></td>
<td>0.91</td>
<td>0.76</td>
<td>0.57</td>
<td>0.89</td>
</tr>
</tbody>
</table>

Key: Prop. = proposed technique [29], Tang = Tang and Hang technique [26], Pal = Pal, Das, Biswas and Mukhopadhyay technique [17].
Chapter 6: Results and Discussions

6.4. Colour image watermarking results

Computer simulations were also performed on various colour images to assess the performance of the watermarking techniques. Since majority of the test images are in the RGB colour space, it was necessary to convert to other colour spaces to assess the performance in those spaces. The performance of the watermarking algorithm in different colour spaces is presented next.

i. RGB colour space

The RGB images were split into their component images, Red, Green and Blue components, and then the watermark was embedded using the algorithm described in section 5.5. The watermark was embedded in each component image, one at a time, to test which one gave the best trade-off between robustness and perceptibility.

a) Visual perceptibility test

Figure 6.16 shows a comparison of perceptibility assessment when the watermark is embedded into the different component images and the message extracted in each case.

![Peppers image components](image)

Figure 6.16 (a) Red, (b) Green and (c) Blue components of Peppers image having PSNR values of 36.90dB, 39.02dB and 38.74dB respectively and the corresponding extracted watermarks having NC factors of 0.98, 0.99 and 0.99 respectively.
Chapter 6: Results and Discussions

The results given in Figure 6.16 show the performance of the proposed technique on different components of the low frequency *Peppers* image while Figure 6.17 shows its performance on the *Baboon* image which is dominated by high frequency features.

![Baboon image](image)

*Figure 6.17 (a) Red, (b) Green and (c) Blue components of the Baboon image having PSNR values of 43.83dB, 44.13dB and 43.61dB respectively and the corresponding extracted watermarks having NC factors of 0.99.*

The tests that were performed showed no significant disparity on the performance of the watermarking algorithm regardless of the component image chosen for the embedding process.

To assess the effectiveness of the proposed technique on colour images, the RGB images were subjected to a variety of signal processing and geometric attacks.

**b) Signal processing attack tests**

Figure 6.18 shows the *Couple* and *House* images subjected to Gaussian noise attack with zero mean and a variance of 0.02, and the recovered watermarks. These are images dominated by man-made structures such as buildings which exhibit sharp edges and low frequency components.
Chapter 6: Results and Discussions

Figure 6.18 The (a) Couple and (b) House images subjected to Gaussian noise attack with a mean of zero and variance of 0.02 and their respective extracted watermark.

From Figure 6.18 it can be deduced that the embedding technique is robust to Gaussian noise attack. In other simulation tests the results showed robustness to signal processing attacks including JPEG compression, histogram equalisation and filtering attacks.

c) RST attack tests

The colour images were subjected to a variety of geometric distortion attacks including RST and cropping attacks in order to assess the performance of the proposed technique. Figures 6.19 and 6.20 show the Delaunay tessellations used to restore an image that has been subjected to rotation attack. Figures 6.21 and 6.22 show the selected probe and target triangles used in restoring the attacked images in Figures 6.19 and 6.20 respectively to their original positions before the watermarks are extracted.
Figure 6.19 Delaunay triangulation of (a) the cover and (b) the attacked Girl image.

Figure 6.20 Delaunay triangulation of (a) the cover and (b) the attacked House image.
Chapter 6: Results and Discussions

Figure 6.21 (a) Target and (b) probe triangles at RF of 39° and (c) the extracted watermark at NC of 0.98.

Figure 6.22 (a) Target and (b) probe triangles of the House image at RF of 129° and (c) the extracted watermark at NC of 0.98.

The proposed technique is adequately robust to rotation attack on RGB colour images as is evident in Figures 6.21 and 6.22. Simulations were also performed on scale, translation and cropping attacks and the results obtained summarised in Table 6.2. The results given indicate the NC value of the extracted watermarks.
Table 6.2 Summary of performance of the proposed technique to affine geometric attacks on RGB colour images.

<table>
<thead>
<tr>
<th>Attack Method</th>
<th>Rotation</th>
<th>Scale 50%</th>
<th>Scale 75%</th>
<th>Scale 125%</th>
<th>Translation 5 pixels right</th>
<th>Translation 10 pixels down</th>
<th>RST (32,1.25,14)</th>
<th>Cropping 10%</th>
<th>Cropping 20%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Image</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>House</td>
<td>0.98</td>
<td>0.88</td>
<td>0.94</td>
<td>0.92</td>
<td>0.95</td>
<td>0.93</td>
<td>0.85</td>
<td>0.95</td>
<td>0.87</td>
</tr>
<tr>
<td>Girl</td>
<td>0.97</td>
<td>0.91</td>
<td>0.95</td>
<td>0.92</td>
<td>0.96</td>
<td>0.97</td>
<td>0.84</td>
<td>0.95</td>
<td>0.82</td>
</tr>
<tr>
<td>Baboon</td>
<td>0.96</td>
<td>0.91</td>
<td>0.94</td>
<td>0.93</td>
<td>0.98</td>
<td>0.97</td>
<td>0.89</td>
<td>0.97</td>
<td>0.93</td>
</tr>
<tr>
<td>Peppers</td>
<td>0.94</td>
<td>0.92</td>
<td>0.97</td>
<td>0.96</td>
<td>0.96</td>
<td>0.97</td>
<td>0.88</td>
<td>0.96</td>
<td>0.87</td>
</tr>
<tr>
<td>Couple</td>
<td>0.96</td>
<td>0.94</td>
<td>0.97</td>
<td>0.95</td>
<td>0.96</td>
<td>0.97</td>
<td>0.96</td>
<td>0.97</td>
<td>0.90</td>
</tr>
</tbody>
</table>
Chapter 6: Results and Discussions

From Table 6.2 it is evident that the proposed watermarking technique has sufficient robustness to most affine geometric attacks on RGB colour images.

ii. CMY colour space

Just like the case of RGB colour space, the component images of CMY colour space contain both the intensity and colour information. As a result the choice of the embedding component did not affect the robustness or perceptibility of the watermark significantly as evidenced in Figure 6.23. When CMY colour images were subjected to signal processing attacks and geometric attacks the results were found to mirror the results obtained in the RGB colour images. It was therefore safe to conclude that the proposed watermarking technique is robust when implemented on images in the CMY colour space.

![CMY Colour Components](image)

*Figure 6.23 (a) Cyan, (b) Magenta and (c) Yellow components of the Peppers image having PSNR values of 37.43dB, 38.23dB and 36.61dB respectively and the corresponding extracted watermarks having NC factors of 0.99, 0.98 and 0.99 respectively.*

iii. NTSC colour space

The colour test images were converted to NTSC colour space and the watermark embedded in the luminance (Y) and chrominance (I and Q) component images and their performances compared in Figure 6.24 and 6.25, for Peppers and Baboon images.
Chapter 6: Results and Discussions

(a) Luminance, (b) Hue and (c) Saturation components of Peppers image having PSNR values of 40.03dB, 31.28dB and 29.84dB and their corresponding extracted watermark having NC factors of 0.99, 0.93 and 0.89 respectively.

The NTSC image was also subjected to RST attacks and the proposed technique used to reverse the attacks before attempting to retrieve the embedded watermark. The results obtained from simulating these attacks on the Peppers and the Baboon images in the NTSC colour space are given in Figures 6.26 and 6.27.

(a) Luminance, (b) Hue and (c) Saturation components of Baboon image having PSNR values of 43.92dB, 39.67dB and 37.43dB and their corresponding extracted watermark having NC factors of 0.99, 0.95 and 0.90 respectively.
Figure 6.26 (a) The probe and (b) target triangles of Peppers image in NTSC colour space and the retrieved watermark at an NC factor of 0.99.

From the results illustrated in Figures 6.24, 6.25, 6.26 and 6.27 it can be seen that the luminance component of the NTSC colour space provides the best platform for embedding the watermark. It is also evident that the proposed technique can be effectively employed in this colour space to attain robustness of a watermark to geometric attacks. The results obtained after geometric attacks on the NTSC colour space is used as a representative to illustrate the proposed techniques invariance to geometric attacks when used on images which have specific components for intensity and others for colour information.

iv. HSV and HSI colour spaces

The test images were converted from RGB colour space to the HSV and HSI colour spaces and the performance and effect of watermarking each component image at a time evaluated. Figures 6.28 and 6.29 show the results obtained from watermarking the Hue, Saturation and Value component images of the House and Girl test images.
Chapter 6: Results and Discussions

Figure 6.27 (a) The probe and (b) target triangles of the Baboon in NTSC colour space and the retrieved watermark.

Figure 6.28 Hue (a), Saturation (b) and Value (c) component images of the House and the corresponding watermark extracted from each. Their respective PSNR and NC factors are 24.53dB, 28.48dB and 36.85dB and 0.89, 0.94 and 0.99 respectively.

From Figures 6.28 and 6.29 it is evident that the proposed technique can be used to embed watermark into an image with minimal quality degradation when the embedding is done in the value component of an HSV image.
Chapter 6: Results and Discussions

Figure 6.29 Hue (a), Saturation (b) and Value (c) component images of the Girl and the corresponding watermark extracted from each. Their respective PSNR and NC factors are 29.45 dB, 31.67 dB and 41.43 dB and 0.91, 0.95 and 0.99 respectively.

v. YCbCr colour space

The results obtained from watermarking the component images of the YCrCb colour space are illustrated in Figures 6.30 and 6.31.

Figure 6.30 Component images of the Girl in YCbCr colour space and the corresponding watermark extracted from each. They have PSNR values of 34.28 dB, 20.43 dB and 23.64 dB and NC factors of 0.98, 0.92 and 0.94 respectively.
Chapter 6: Results and Discussions

Figure 6.31 Component images of the Girl in YCbCr colour space and the corresponding watermark extracted from each. They have PSNR values of 34.28dB, 20.43dB and 23.64dB and NC factors of 0.97, 0.92 and 0.94 respectively.

From the results obtained in Figures 6.30 and 6.31 it is evident that embedding the watermark in the luminance component Y produces the best result. Like in the previous colour spaces it is observed that the proposed technique can be used to embed watermarks in the YCbCr colour space giving satisfactory results.

6.5 Video watermarking

The test videos were embedded following the procedure outlined in section 5.9. Since the videos were in YUV or YCbCr format, the luminance component of the pictures or frames was used in the embedding process as it gave the best results in this colour space. Various embedding algorithms were simulated to test their robustness to the various video coding attacks and the results given below.

Figure 6.32 shows the results obtained from embedding the watermark in the frames of Foreman after testing its robustness to compression by coding it with the H.264 standard. The video is of CIF format of resolution 352 × 288 pixels and pixel depth of 24 bits per pixel.
Chapter 6: Results and Discussions

Figures 6.33 and 6.34 show the cover and watermarked frames of the *Bus* and *Container* test videos and the extracted watermarks after the videos have been compressed using the H.264 coding standard.

![Figure 6.32 Cover, (a) watermarked (b) frames of the Foreman and the extracted message (c) having a PSNR value of 29.74 dB and NC factor of 0.98.](image)

![Figure 6.33 Cover (a), watermarked (b) frames of the Bus and the extracted message (c) after H.264 coding. The watermarked frame has a PSNR value of 37.97 dB and NC factor of 0.99.](image)

The video watermarking results illustrated in Figures 6.31, 6.32 and 6.33 demonstrate that the technique applied was effectively used to embed the watermark to frames of a video with minimal visual degradation. It is also evident that the watermarking technique showed robustness to H.264 compression standard. Table 6.3 illustrates the performance of various video watermarking schemes.
Chapter 6: Results and Discussions

Figure 6.34 Cover (a), watermarked (b) frames of the Container and the extracted message (c) after H.264 coding. The watermarked frame has a PSNR value of 42.89 dB and NC factor of 0.99.

Table 6.3 Comparing the performances of video watermarking techniques.

<table>
<thead>
<tr>
<th>Performance Parameter</th>
<th>Visual Perceptibility (Frame Level)</th>
<th>H.264 Compression</th>
<th>Inter-frame Interference</th>
<th>Computational Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>Pass</td>
<td>Pass</td>
<td>Fail</td>
<td>Pass</td>
</tr>
<tr>
<td>Polyak</td>
<td>Pass</td>
<td>Fail</td>
<td>Fail</td>
<td>Pass</td>
</tr>
<tr>
<td>Zhang</td>
<td>Pass</td>
<td>Pass</td>
<td>Pass</td>
<td>Fail</td>
</tr>
</tbody>
</table>

From Table 6.3 it is evident that the proposed scheme still suffered from inter-frame interference which reduced the quality of the video after the frames were compiled together. Further work is still being carried out to improve the video quality of the scheme by reducing inter-frame interference.
7.1. Conclusions

i. Image watermarking

In this thesis it has been demonstrated that VQ and Delaunay tessellation techniques can be used to improve the robustness of a watermarked image to geometric attacks. The results in the previous chapter have shown that the watermark can be recovered if the cover image has been subjected to small angles of rotation and cropping out small portions of the image using the technique proposed by Pal et al [17] which employs VQ techniques. The results have also demonstrated that the watermark can be recovered if the cover image is subjected to certain angles of rotation, ratios of scaling and translations using the technique employed by Tang and Hang [26] which employs the use of Delaunay tessellation techniques.

This thesis has exploited the robustness exhibited by both techniques and further improving on the technique used by Tang and Hang [26] to develop a technique that is robust to RST and cropping attacks. In the proposed technique the watermark is extracted at higher NC factors even when the angles of rotation and ratios of scaling lead to loss of reference points.

It can therefore be concluded that the proposed technique was successfully simulated in MATLAB platform and showed an improved robustness to geometric distortion attacks while maintaining an excellent perceptual quality of the watermarked image. The technique also showed superior robustness in comparison to other related techniques.

The technique however suffered from lack of capacity and was unable to hold larger messages, as a cover image of 256 × 256 pixels could hold a message with a maximum of 1024 bits if all the vectors obtained by the vectorisation process are maintained in the codebook and all blocks
used in the embedding process. In cases where compression of the image demands use of fewer codewords, the capacity of the embedded watermark is then further reduced.

ii. **Video watermarking**

In addition to the challenges encountered in image watermarking, video watermarking is vulnerable to numerous other attacks therefore requiring modification in the technique used. The technique that was used exhibited robustness to the video coding challenge when the video sequences were subjected to H.264 compression standard as evidenced from the results obtained.

7.2. **Recommendations**

i. **Image watermarking**

The process of codebook design and optimization was time consuming hence limiting the possibility of employing this technique in real-time applications such as broadcast monitoring. It is therefore not recommended to include the VQ techniques when watermarking real-time applications. However, this slightly reduces the robustness of the watermark to some geometric attacks.

The capacity of the embedding also depended on the size of the codebook as only one bit of the watermark can be embedded in each block as explained in section 3.3, therefore the compression ratio affected the capacity of the watermark. Prediction error expansion techniques have been used to improve the capacity in the spatial domain as proposed by Li et al [5]. Further work on the possibility of using prediction error expansion techniques in the transform domain is recommended in order to improve the watermarking capacity in the transform domain. Further work is also being done on watermarking fingerprints and also on using fingerprints to watermark documents.
ii. Video watermarking

In the video watermarking process, further work is recommended on embedding of the watermark on visually important areas of the frames of uncompressed video as proposed by Hsieh and Lin [42]. If this technique is done with little quality degradation, it would solve the problem of inter-frame distortion drift while making the watermark robust to all compression standards.

Further work is also recommended in temporal and spatial collision attacks which are used in detecting the presence of watermark sequence in frames. Further work is also recommended on alternative ways of reducing inter-frame distortion drift while not increasing computational complexity.
REFERENCES


APPENDIX A: MATLAB PROGRAMS

This appendix contains the programs that were used for simulating the embedding and extraction process of the proposed techniques. The programs are arranged in the order that they are simulated beginning with the embedding process and finally the extraction process. Some of the programs used were written by the author while others were obtained from authors who have done related work and then modified to suit the particular requirements of the study. Some programs were also obtained from the MathWorks website [46].

a) Programs used in the embedding process

This section gives the programs used during the process of embedding the watermark; they include the programs used in the VQ process and DCT embedding procedure.

```matlab
% This program is used to encode the image into codebook for watermark
% embedding using the VQ process
% Name: Felix Owalla
% Institution: University of Nairobi
% Date: April 2011.

x = imread('peppers.bmp');
original = x;

cb = 1024; % Set Codebook size
s = 8; % Set block size
x = double(x);

rowsz = size(x,1); % Number of rows of image
colsz = size(x,2); % Number of columns of image
rem1 = mod(rowsz, s); % Check if number of rows are divisible by 8
rem2 = mod(colsz, s); % Check if number of columns are divisible by 8

% Pad the image with zeros if rows or columns are not divisible by 8
if rem1 ~= 0
    r = s - rem1;
elser = 0;
end
```

94
Appendix A: MATLAB Programs

```matlab
if rem2~=0
c=s-rem2;
else
c=0;
end

x=padarray(x,[r c]);

%% Vectorise the image
x1=im2col(x,[s s],'distinct');
x1=x1';

%%initialize codebook

rndcodb=rand(c*b,s*s);
rndcodb=rndcodb*255;
imwrite(uint8(rndcodb),'codebook.bmp');
rndcodb =imread('codebook.bmp');
rndcodb =double(rndcodb);

cond=0;
N1=1;
if(cond~=0)
N1=cond;
N=10;
end

for i=N1:N
    eta=etaz*(1-i/N);
    for j=1:cb
        mini=(((repmat(x(j,:),[c b 1])-rndcodb(:,:)).^2))';
        if(not (size(x,2)==1))
            mini=(sum(mini))';
        end
        mini=sqrt(mini);
        m=min(mini);
        temp=find(mini==m);
        if(length(temp)==1)
            win=temp;
        else
            win=temp(1);
        end
        codb=rndcodb(:,1:s*s);
        if(p==0)
imwrite(uint8(codb),'codebook.bmp');
        compressed=codb;
```
for i=1:1
if(i==1)
codb=imread('codebook.bmp');
s1='Reconstructed';
else
end

c=size(codb,1);

% Find nearest code
for i=1:n
mini=(((repmat(x(i,:),[c 1])-codb(:,:,).^2))');
if(not (size(x,2)==1))
mini=(sum(mini))';
end
mini=sqrt(mini);

m=min(mini);
temp=find(mini==m);
if(length(temp)==1)
index(i)=temp;
else
index(i)=temp(1);
end
clear temp;

imag=reshape(index,nx/s,ny/s);

Algorithm A.1. VQ encoding process.

% This program is used to embed watermark in the DCT domain
% Name: Chris Shoemaker [15]
%Project: DCT mid-band Watermark Embedding Using PN sequences

clear all;

start_time=cputime; % save start time
k=20; % set gain factor for embedding
blocksize=8; % set the dct blocksize

midband=[ 0,0,0,1,1,1,1,0; % defines the mid-band frequencies of an 8x8 dct
0,0,1,1,1,1,0,0;
0,1,1,1,1,0,0,0;
1,1,1,1,0,0,0,0;
1,1,1,0,0,0,0,0;
1,1,0,0,0,0,0,0;
1,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0 ];

% read in the cover object
Appendix A: MATLAB Programs

file_name='lena_std_bw.bmp';
cover_object=double(imread(file_name));

% determine size of cover image
Mc=size(cover_object,1); %Height
Nc=size(cover_object,2); %Width

% determine maximum message size based on cover object, and blocksize
max_message=Mc*Nc/(blocksize^2);

% read in the message image
file_name='copyright.bmp';
message=double(imread(file_name));
Mm=size(message,1); %Height
Nm=size(message,2); %Width

% reshape the message to a vector
message=round(reshape(message,Mm*Nm,1)./256);

% check that the message isn't too large for cover
if (length(message) > max_message)
    error('Message too large to fit in Cover Object')
end

% pad the message out to the maximum message size with ones's
message_vector=ones(1,max_message);
message_vector(1:length(message))=message;

% generate shell of watermarked image
watermarked_image=cover_object;

% read in key for PN generator
file_name='_key.bmp';
key=double(imread(file_name))./256;
rand('state',key); % reset MATLAB's PN generator to state "key"

% generate PN sequence
pn_sequence_zero=round(2*(rand(1,sum(sum(midband)))-0.5));

% process the image in blocks
x=1;
y=1;
for (kk = 1:length(message_vector))
    % transform block using DCT
    dct_block=dct2(cover_object(y:y+blocksize-1,x:x+blocksize-1));
    % if message bit contains zero then embed pn_sequence_zero into the mid-band
Appendix A: MATLAB Programs

% components of the dct_block
ll=1;
if (message_vector(kk)==0)
    for ii=1:blocksize
        for jj=1:blocksize
            if (midband(jj,ii)==1)
                dct_block(jj,ii)=dct_block(jj,ii)+k*pn_sequence_zero(ll);
                ll=ll+1;
            end
        end
    end
end
end
end

% transform block back into spatial domain
watermarked_image(y:y+blocksize-1,x:x+blocksize-1)=idct2(dct_block);

% move on to next block. At and of row move to next row
if (x+blocksize) >= Nc
    x=1;
    y=y+blocksize;
else
    x=x+blocksize;
end
end

% convert to uint8 and write the watermarked image out to a file
watermarked_image_int=uint8(watermarked_image);
imwrite(watermarked_image_int,'dct2_watermarked.bmp','bmp');

elapsed_time=cputime-start_time,    % display processing time

% display psnr of watermarked image
psnr=psnr(cover_object,watermarked_image,Nc,Mc),

% display watermarked image
figure(1)
imshow(watermarked_image,[])

Algorithm A.2. Watermark embedding process. Courtesy of Chris Shoemaker [17].

% This program is used to decode the watermarked image from the codebook
% embedded with the message
% Name: Felix Owalla
% Institution: University of Nairobi
% Date: April 2011.

imag=imread('codebook.bmp');
Appendix A: MATLAB Programs

```matlab
x=double(imag);
cond=0;
N1=1;
if(cond~=0)
    N1=cond;
    N=10;
end

for i=N1:N
    eta=etaz*(1-i/N);
    for col=1:cb
        mini=((repmat(x(col,:),[cond 1])-x(:,:)).^2)';
        if(not (size(x,2)==1))
            mini=(sum(mini))';
        end
        mini=sqrt(mini);
        m=min(mini);
        temp=find(mini==m);
        if(length(temp)==1)
            win=temp;
        else
            win=temp(1);
        end
        if=mini;
        alpha=1;
        beta=1;
        gama=0.5000;
        vnew=v(win,:)+eta*(x(j,:)-v(win,:)).*(1+sum(w));
        for k=1:c
            v(k,:)=v(k,:)+eta*(x(j,:)-v(k,:)).*nw(k);
        end
        codb(win,:)=codbnew;
    clear temp;
end
codb=codb(:,1:s*s);
t=codb(x(:,:))';
image=col2im(t,[s s],[nx*s ny*s],'distinct');
```

Algorithm A.3. VQ decoding process.
Appendix A: MATLAB Programs

b) Programs used in the watermark extraction process

In this subsection the programs used in the watermark recovery process are presented. They include the programs used for the restoration of attacked images and watermark extraction from the DCT domain. Though VQ encoding and decoding are also used during the watermark extraction process, they are not presented in this subsection as they are the same ones used in the embedding process.

% This program is used to extract the feature points and to obtain % the Delaunay tessellation of the attacked images % Name: A. Ganoun % Institution: The MathWorks Inc [46].

frame=imread('cameraman.tif');
I =double(frame);
imshow(frame);
k = waitforbuttonpress;
point1 = get(gca,'CurrentPoint');   %button down detected
rectregion = rbbox; %return figure units
point2 = get(gca,'CurrentPoint'); %button up detected
point1 = point1(1,1:2); % extract col/row min and maxs
point2 = point2(1,1:2);
lowerleft = min(point1, point2);
upperright = max(point1, point2);
ymin = round(lowerleft(1));
ymax = round(upperright(1));
xmin = round(lowerleft(2));
xmax = round(upperright(2));
% Interest Points
Aj=6;
cmin=xmin-Aj; cmax=xmax+Aj; rmin=ymin-Aj; rmax=ymax+Aj;
min_N=12;max_N=16;
% The Mask
sigma=2; Threshold=20; r=6; disp=1;
dx = [-1 0 1; -1 0 1; -1 0 1];
dy = dx';
% Gaussian Filter
Ix = conv2(I(cmin:cmax,rmin:rmax), dx, 'same');
Iy = conv2(I(cmin:cmax,rmin:rmax), dy, 'same');
g = fspecial('gaussian',max(1,fix(6*sigma)), sigma);

Ix2 = conv2(Ix.^2, g, 'same');
Iy2 = conv2(Iy.^2, g, 'same');
Ix y = conv2(Ix.*Iy, g, 'same');
Appendix A: MATLAB Programs

```matlab
k = 0.04;
R11 = (Ix2.*Iy2 - Ixy.^2) - k*(Ix2 + Iy2).^2;
R11=(1000/max(max(R11)))*R11;
R=R11;
ma=max(max(R));
sze = 2*r+1;
MX = ordfilt2(R,sze^2,ones(sze));
R11 = (R==MX) & (R>Threshold);
count=sum(sum(R11(5:size(R11,1)-5,5:size(R11,2)-5)));

loop=0;
while (((count<min_N)|(count>max_N))&(loop<30))
if count>max_N
    Threshold=Threshold*1.5;
elseif count < min_N
    Threshold=Threshold*0.5;
end

R11 = (R==MX) & (R>Threshold);
count=sum(sum(R11(5:size(R11,1)-5,5:size(R11,2)-5)));
loop=loop+1;
end

R=R*0;
R(5:size(R11,1)-5,5:size(R11,2)-5)=R11(5:size(R11,1)-5,5:size(R11,2)-5);
[r1,c1] = find(R);
PIP=[r1+cmin,c1+rmin] % IP
y = r1+cmin; x= c1+rmin; % helps me with the tessellation of attacks

Size_PI=size(PIP,1);
for r=1: Size_PI
    I(PIP(r,1)-2:PIP(r,1)+2,PIP(r,2)-2)=255;
    I(PIP(r,1)-2:PIP(r,1)+2,PIP(r,2)+2)=255;
    I(PIP(r,1)-2,PIP(r,2)-2:PIP(r,2)+2)=255;
    I(PIP(r,1)+2,PIP(r,2)-2:PIP(r,2)+2)=255;
end

imshow(uint8(I))
% Display tessellation

tri = delaunay(x,y);
imshow(frame); hold on;
```
Algorithm A.4. Image synchronisation process.

% This program is used to extract the embedded message from the codebook
% Name: Chris Shoemaker
% Project: Watermark Recovery in DCT mid-band

clear all;

% save start time
start_time=cputime;

blocksize=8; %set the dct blocksize

midband=[ 0,0,0,1,1,1,1,0; %defines the mid-band frequencies of an 8x8 dct
         0,0,1,1,1,1,0,0;
         0,1,1,1,1,0,0,0;
         1,1,1,1,0,0,0,0;
         1,1,1,0,0,0,0,0;
         1,1,0,0,0,0,0,0;
         1,0,0,0,0,0,0,0;
         0,0,0,0,0,0,0 ];

% read in the watermarked object
file_name='dct2_watermarked_mod.bmp';
watermarked_image=double(imread(file_name));

% determine size of watermarked image
Mw=size(watermarked_image,1); %Height
Nw=size(watermarked_image,2); %Width

% determine maximum message size based on cover object, and blocksize
max_message=Mw*Nw/(blocksize^2);

% read in original watermark
file_name='_copyright.bmp';
orig_watermark=double(imread(file_name));

Mo=size(orig_watermark,1); %Height
No=size(orig_watermark,2); %Width

% read in key for PN generator
file_name='_key.bmp';
key=double(imread(file_name))./256;

rand('state',key); % reset MATLAB's PN generator to state "key"
% generate PN sequence
Appendix A: MATLAB Programs

pn_sequence_zero = round(2*(rand(1,sum(sum(midband))) - 0.5));

% process the image in blocks
x=1;
y=1;
for (kk = 1:max_message)

% transform block using DCT
    dct_block = dct2(watermarked_image(y:y+blocksize-1,x:x+blocksize-1));

% extract the middle band coefficients
    ll=1;
    for ii=1:blocksize
        for jj=1:blocksize
            if (midband(jj,ii)==1)
                sequence(ll)=dct_block(jj,ii);
                ll=ll+1;
            end
        end
    end

% calculate the correlation of the middle band sequence to pn sequence
    correlation(kk)=corr2(pn_sequence_zero,sequence);

% move on to next block. At and of row move to next row
    if (x+blocksize) >= Nw
        x=1;
        y=y+blocksize;
    else
        x=x+blocksize;
    end
end

% if correlation exceeds threshold, set bit to '0', otherwise '1'
for (kk=1:Mo*No)
    if (correlation(kk) < mean(correlation(1:Mo*No)))
        message_vector(kk)=0;
    else
        message_vector(kk)=1;
    end
end

% reshape the embeded message
message=reshape(message_vector(1:Mo*No),Mo,No);

elapsed_time=cputime - start_time, % display processing time

% display recovered message
figure(2), imshow(message,[],)
APPENDIX B: PUBLICATION 1

This appendix contains the paper that was published by IEEE Xplore in the proceedings of the 16th IEEE Mediterranean Electrotechnical Conference (MELECON 2012) which was held in Yasmine-Hammamet, Tunisia from 25th to 28th March 2012. The results that were obtained in the course of this research were presented and disseminated at the conference ahead of the publication. The image below is taken from the homepage of the MELECON 2012 website.

Figure B.1 Homepage of the IEEE MELECON 2012 (www.melecon2012.com).
A Robust Image Watermarking Scheme Invariant to Rotation, Scaling and Translation Attacks

Felix O. Owalla, Student Member, IEEE and Elijah Mwangi, Member, IEEE
Department of Electrical and Information Engineering
University of Nairobi
P.O. Box 30197 – 00100
Nairobi, Kenya.
f.owalla@students.uonbi.ac.ke

Abstract— A robust digital image watermarking scheme that is resistant to both signal processing and geometric attacks is proposed. The watermark is embedded in the Discrete Cosine Transform domain in a spread-spectrum format and Vector Quantization techniques used to compress the image. The recovery of the watermark after rotation, scaling and translation attacks is done by using Harris corner detector-based feature-points to get a Delaunay tessellation which is used to reverse the attacks. In situations where RST attacks lead to formation of substantial dark areas on the image, some reference feature-points are lost and recovered watermark is poor or entirely lost. In our proposed scheme, a procedure of estimating the RST attacks is employed by taking an average of selected triangles in the tessellation. Computer simulation results using MATLAB have been used to confirm the accuracy of our proposed scheme.

Index Terms: Delaunay triangulation, Spread-spectrum and Vector Quantization.

I. INTRODUCTION

Watermarking has been used extensively to protect digital media from illegal copying and reproduction. However, this is equally threatened by attackers who use a variety of attacks to remove or to render the watermark useless. These attacks can be roughly grouped into signal processing attacks or geometric attacks. Geometric attacks are difficult to deal with as they involve displacement of pixels thereby inducing synchronization errors between the original and extracted watermarks during detection process [1].

The watermark should also be imperceptible and should not degrade the quality of the image. In order to attain this, the watermark can be embedded in a domain such as the Discrete Cosine Transform (DCT), using the mid frequencies which do not contain visually important features of the image and hence largely unaffected by filtering and noise attacks. The DCT is sufficiently robust to signal processing attacks but very fragile to geometric attacks [1], [2]. There are a few geometric-distortion focused watermarking schemes; they can be roughly grouped into: moment-based, template-based, invariant domain-based and feature-point-based schemes [2]. Feature-point-based techniques have been reported to offer high resistance to geometric attacks. These feature-point-based techniques include Harris corner detector and Mexican hat wavelet [3], [4]. The Mexican hat wavelet however gives synchronization errors when the geometric attacks are local and therefore Harris corner detector is preferred. The feature-points obtained using Harris corner detector are then used for Delaunay triangulation and Voronoi tessellation which are used for synchronization or generally define the embedding regions [3], [4]. For the images to be efficiently stored and transmitted Vector Quantization (VQ) is often used to compress the image.

Some extracted feature points are lost when the attacks lead to formation of substantial dark regions around the image and hence giving a different Delaunay tessellation. The difference in the tessellation leads to synchronization errors as it is more difficult to estimate the rotation, scaling and translation (RST) attack. Our proposed scheme uses averaging of triangles to estimate the exact nature of attack on the image in order to restore it to the position of the original image. This ensures that the watermark can still be extracted even after losing a number of reference points.

This paper is organized as follows. Section (II) describes the extraction of feature points. Section (III) describes Delaunay triangulation. Section (IV) describes the watermark embedding process, the watermark recovery procedure and the proposed scheme presented. Section (V) presents the experimental results and the conclusion given in Section (VI).

II. FEATURE-POINTS EXTRACTION

After reconstructing the image from the watermarked codebook of the image, the feature-points which will be used in synchronizing the attacked image are obtained [4]. Robust feature-points are obtained using Harris corner detector. This is done by obtaining feature-points and then rotating the image by various angles and repeating the above process, the features which are extracted for all the angles are then taken as the robust feature-points.

Harris corner detector is based on local auto-correlation function of a signal. Local auto-correlation
functions measure the local changes of the signal with patches shifted by a small amount in different directions. Given a shift \((\Delta x, \Delta y)\) and a point \((x, y)\), the autocorrelation function is defined as

\[ c(x, y) = [\Delta x \Delta y] C(x, y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}. \] (1)

Where matrix \(C(x,y)\) is given by

\[ C(x,y) = \begin{bmatrix} \sum_w I_x(x, y) \sum_w I_y(x, y) I_x(x, y) I_y(x, y) \\ \sum_w I_x(x, y) I_y(x, y) \sum_w I_y(x, y) \end{bmatrix}. \] (2)

Where \(w\) is a Gaussian window of say 3-by-3 or 5-by-5 pixels centered at \((x,y)\) and \(I_x(x,y)\) and \(I_y(x,y)\) are partial derivatives of the image function \(I(x,y)\) in \(x\) and \(y\) axes respectively.

The matrix \(C(x,y)\) captures the intensity structure of the local neighbourhood. The eigenvalues of the matrix are used to obtain the local auto-correlation function of the windowed image region. When both the eigenvalues are low, the windowed region is flat, when one eigenvalue is high and the other one low the region is an edge and when both are high then the region is a corner.

The feature-points are then used in obtaining Delaunay triangulation. The triangles are employed in synchronizing the image during watermark extraction [3], [5].

III. DELAUNAY TRIANGULATION

The Delaunay tessellation of the feature-points extracted from an image is obtained by determining the nearest neighbours [3]. In order to get the Delaunay tessellation of the feature points, a Voronoi diagram of the image is obtained first.

Voronoi tessellation is constructed by defining the area \(D_i\) consisting of all locations in the space which are closest to a point \(p_i\) than to any other point on the image. The boundaries separating two adjacent Voronoi regions \(D_i\) and \(D_j\) containing points \(p_i\) and \(p_j\) is a perpendicular bisector of the two points.

Fig. 1 illustrates the Voronoi and Delaunay tessellations of a set of points.

To construct the Delaunay tessellation nearest neighbouring points, whose cells in the Voronoi tessellations of a set of feature points of images are illustrated in fig. 2.

Delaunay tessellation is a dual of Voronoi tessellation and employs the concept of an empty circumcircle. The empty circumcircle criterion is such that for any three points forming a Delaunay triangle lie on the circumcircle and no point should lie inside the circumcircle. The empty circumcircle criterion gives Delaunay the properties of maximizing the minimum angle of the triangles and minimizing the size of the circumcircle. Delaunay tessellation also has the property of being unique except in cases of dual circumcircle, which is four points laying on a circumcircle.

IV. PROPOSED METHOD

In this paper an algorithm which combines VQ, DCT and Spread-spectrum in the watermark embedding process and incorporates Delaunay triangulation in the extraction process. The embedding procedure can be summarized as follows. The LBG algorithm is used to design and obtain the codebook of the encoded image, the codebook is again decomposed into 8-by-8 pixel blocks [7], - [9] and the DCT coefficients of each block obtained. The DCT coefficients of the watermarks are then embedded into the coefficients of the codebook in a spread-spectrum format.

The watermark extraction process can be summarized as follows. Delaunay triangulation of the feature points of the image is used to resynchronize the image to the position of the original image. After resynchronization, the codebook is obtained and the DCT coefficients obtained and the watermark extracted from the embedded positions.

A. Embedding Procedure

1. **Step 1.** The LBG algorithm to construct a codebook of the image with a codeword size of 8-by-8.

2. **Step 2.** Decompose the image into 8-by-8 blocks of the codebook and compute a block-based DCT of the codebook and select the coefficients of the mid-frequency sub-band.
Step 3. Generate a pseudo-random sequence and select the value of $\beta$ which controls the embedding strength.

4. **Step 4.** Embed the watermark into the selected mid-frequency sub-band using the pseudo-random (PN) sequence to determine the coefficients where the watermark is embedded as follows:

\[
I^w(u,v) = I(u,v) + \beta W(u,v)
\]

Where, $I(u,v)$, $I^w(u,v)$ and $W(u,v)$ are the original, watermarked and watermark DCT coefficients of the codebook respectively.

5. **Step 5.** Compute the inverse DCT of the watermarked codebook and then perform VQ decoding to obtain the watermarked image.

The value of $\beta$ between 0.1 and 0.5 was found to give a fair trade-off between robustness and visual quality depending on the pixel texture of the image [10]. It was observed that the generally smooth pixel images a higher value of $\beta$ could be used with less visual quality distortion than the rough pixel images.

### B. Watermark Recovery Procedure

The process of watermark recovery from the image is given in this subsection as follows:

1. **Step 1.** Obtain the feature-points of the image using the Harris corner detector then obtain the Delaunay triangulation of the feature-points obtained.

2. **Step 2.** Select several target triangles and corresponding probe triangles from the Delaunay of the original and attacked image respectively.

3. **Step 3.** Using the triangles estimate the amount of distortion that each probe triangle has undergone by comparing the orientation angles, sizes and positions to the corresponding target triangle.

4. **Step 4.** Obtain the average of the distortions undergone by each triangle and use the averages as the rotation factor ($RF$), scaling factor ($SF$) and translation factor ($TF$) which are then used to restore the image to its original form.

5. **Step 5.** Obtain the codebook of the resynchronized image and compute its DCT.

6. **Step 6.** Select the mid-band frequencies and using the same PN sequence used in the embedding process check for the presence or absence of the embedded watermark using correlating the coefficients of the mid-band frequencies and the PN sequence.

7. **Step 7.** Reconstruct the watermark using the extracted watermark bits.

### V. EXPERIMENTAL RESULTS

To evaluate the performance of our proposed scheme, several experiments were performed to determine the visual perceptibility and robustness of the embedded watermark.

#### A. Visual Perceptibility

The watermarked images were assessed for visual distortion and then Peak Signal-to-Noise Ratio (PSNR) used to determine quality degradation as a result of the embedded watermark. For example, for the cameraman test image there is no visible visual degradation in quality at a PSNR value of 34.9 dB. Similar observations were noted for other test images.

#### B. Robustness to Attacks

The watermarked image was subjected to a variety of attacks, including signal processing attacks and geometric distortion attacks. The watermark was then extracted after restorations where necessary and the quality of the watermark computer using a Normalized cross-correlation (NC) factor. The first attack to be simulated was the rotation attack. Several rotation attack angles were simulated and then a reversal was done for each angle and the watermark extracted. Fig. 3 illustrates the probe and target triangle with an $RF$ of 79°. The embedded watermark and the watermark that was recovered after restoring the attacked image to its original position are shown in fig. 4. The recovered message has an NC of 0.99.

![Fig. 3](image-url)

**Fig. 3** The (a) probe triangle and (b) target triangle after a huge rotation distortion $RF = 79^\circ$.

![Fig. 4](image-url)

**Fig. 4** Watermark embedded in cameraman image (a) original message and (b) extracted message after restoring rotation attack.

The second attack to be simulated was the scaling attack. The image was scaled from 25% to 125% of the original image size. At 25% of the original image size the recovered watermark was unsatisfactory, while scaling levels above 50% the watermark is recovered.
Appendix B: Publication 1

successfully after restoration. Fig. 5 shows the tessellation, probe triangle and the watermark recovered from the attacked image at an SF of 0.5.

Fig. 5 (a) Delaunay triangulation, (b) the probe triangle and (c) the recovered message at SF = 0.5

Translation attacks for various TF were also simulated and the watermark was recovered satisfactorily up to a TF of 48 pixels after which it was not recognizable. Fig. 6 illustrates the Delaunay tessellation and probe triangle after translating the image by a TF of (25,0). The recovered message is shown in fig. 6 (c).

Fig. 6 (a) Delaunay triangulation and (b) the probe triangle TF = 25.0 pixels (c) Recovered message for the translation attack

Finally combinations of two or all the three RST attacks are simulated and the embedded watermarks extracted after restoring the image to its original position. Fig. 7 illustrates an image subjected to a combination of RST attacks and the retrieved watermark after a combination of scaling and rotation, rotation and translation and RST with a 3-element tuple factor of $12^o$, 0.75 and (20,5).

Fig. 7(a) Probe triangle after a combination RST attacks and recovered messages for a combination of (b) scaling and rotation, (c) rotation and translation and (d) rotation, scaling and translation attacks

Computer simulation experiments were also conducted with other common test images such as Barbara, Peppers, Lena and Baboon among others. These experiments tested the robustness of the watermark to signal processing attacks such as JPEG compression and histogram equalization.

TABLE I.

<table>
<thead>
<tr>
<th></th>
<th>Cameraman</th>
<th>Lena</th>
<th>Baboon</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPEG (30%)</td>
<td>0.98</td>
<td>0.96</td>
<td>0.98</td>
</tr>
<tr>
<td>JPEG (50%)</td>
<td>0.99</td>
<td>0.98</td>
<td>0.97</td>
</tr>
<tr>
<td>Avg. filt.</td>
<td>0.97</td>
<td>0.97</td>
<td>0.96</td>
</tr>
<tr>
<td>Unsharp filt.</td>
<td>0.98</td>
<td>0.98</td>
<td>0.96</td>
</tr>
<tr>
<td>Histo. equal.</td>
<td>0.97</td>
<td>0.99</td>
<td>0.95</td>
</tr>
<tr>
<td>Cropping 0.1</td>
<td>0.89</td>
<td>0.75</td>
<td>0.93</td>
</tr>
<tr>
<td>Rotation (79°)</td>
<td>0.99</td>
<td>0.89</td>
<td>0.92</td>
</tr>
<tr>
<td>Scaling (0.5)</td>
<td>0.84</td>
<td>0.87</td>
<td>0.91</td>
</tr>
<tr>
<td>Translation(25,0)</td>
<td>0.99</td>
<td>0.86</td>
<td>0.97</td>
</tr>
<tr>
<td>RST</td>
<td>0.91</td>
<td>0.76</td>
<td>0.89</td>
</tr>
</tbody>
</table>

In all situations the watermark was successfully recovered with a high NC factor and the numerical results are shown in table 1. In addition the test images were subjected to RST attacks and our proposed scheme employed to recover the watermark. This was also successful and the results are shown in table 1.

VI. CONCLUSION

In this paper we have demonstrated that a watermark can be recovered after RST attacks on an image by employing Delaunay tessellation techniques. In situations where RST attacks lead to formation of large dark regions, our proposed scheme of averaging has been shown to be effective in recovering the watermark at high NC factors. The scheme has been tested with success on various test images on a MATLAB simulation platform.

REFERENCES


Appendix B: Publication 1


APPENDIX C: PUBLICATION 2

This appendix consists of work that was accepted and presented at the IEEE Africon 2013 Conference which was held from 9th to 12th September 2013 in Mauritius. The paper has been indexed and published in the digital library of the IEEE Xplore under the conference proceedings. Figure C.1 below shows an image taken from the homepage of the IEEE Africon 2013 website.

Fig. C.1 Homepage of the IEEE Africon 2013 (www.africon2013.org)
A Colour Image Watermarking Technique Resistant to Affine Geometric Attacks

Felix O. Owalla, Graduate Student Member, IEEE and Elijah Mwangi, Member, IEEE
Department of Electrical and Information Engineering
University of Nairobi
P.O. Box 30197 – 00100
Nairobi, Kenya.
f.owalla@students.uonbi.ac.ke

Abstract—This paper presents a colour image watermarking scheme resistant to affine geometric distortions. The technique involves embedding of the watermark in the host image using Discrete Cosine Transform (DCT) domain in the spread-spectrum format. Harris corner detector based feature-points are then employed to get Delaunay triangulation which is used to reverse geometric attacks before attempting to extract the watermark. In some instances the geometric attack may lead to loss of feature-points resulting in the recovery of low quality watermark. In this paper we propose a scheme of estimating geometric attacks by taking the mean of selected triangles in the Delaunay tessellation. Computer simulation using MATLAB has been used to show the robustness of the proposed scheme.

Index Terms: Delaunay triangulation, Image watermarking and Vector Quantization.

I. INTRODUCTION

The development of the World Wide Web has led to ease of illegal reproduction and distribution of digital multimedia products. Several techniques including Digital Rights Management (DRM), cryptography, and watermarking have been used to minimise or discourage the illegal production and distribution. Of these three, watermarking has been found to be more effective and computationally simpler in the protection of digital media.

Attackers use a variety of techniques in an attempt to remove or render the watermark useless. The most common attacks on watermarks can be grouped into signal processing attacks and geometric attacks. Embedding the watermark in the mid-frequency sub-bands of the image in the DCT domain gives it sufficient robustness to signal processing attacks. However since geometric attacks involve displacement of pixels, they induce synchronization errors between the original and extracted watermarks during detection process [1]. These induced errors complicate the watermark recovery process and require a reversal of the geometric distortion before the extraction of the watermark is attempted. In order to reverse the geometric distortion, it is necessary to know the exact nature of distortion used in attacking the cover image.

The embedding process in colour images begins by choosing the component on which the watermark is to be embedded, depending on the colour space of the image. It has been reported that the luminance component gives the best results in some certain colour spaces while green component is the best suited for RGB colour space [2].

The watermark is embedded in the DCT domain’s mid-frequency bands in order to attain visual imperceptibility and robustness to signal processing attacks. However this domain is known to be fragile to geometric attacks [1], [3]. Geometric-distortion resilient watermarking schemes are cartegorised into either moment-based, template-based, invariant domain-based or feature-point-based schemes [4].In moment-based watermarking schemes, the cover image is normalised then an invariant moment e.g. Zernike is used to determine the locations where the watermark message is embedded. The moment-based watermarking schemes are computationally complex [5]. Template-based techniques on the other hand employ the use of a template to synchronise an attacked image to its original position before extracting the watermark. This technique is easily detectible and removable hence it does not attain robustness required [4]. The invariant domain-based schemes employ domains that are resilient to rotation, scale and translation attacks such as the Fourier-Mellin transform. These schemes however suffer from computation inefficiencies and complexities [4], [5].

Feature-point-based techniques involve extraction of reference points from the cover image which are then used to get Delaunay triangulation which is in turn used in synchronising the attacked image to its original position. These feature-point-based techniques include Harris corner detector and Mexican hat wavelet [6], [7]. The Mexican hat wavelet suffers from synchronization errors when the geometric attacks are local and therefore Harris corner detector is preferred. In this paper we use
Appendix C: Publication 2

Feature-points obtained using Harris corner detector which are then used for Delaunay triangulation which is used for synchronization. Vector Quantization (VQ) is used to compress the image and improve on the robustness of the embedding process [3].

Some angles of rotation, ratios of scaling and amount of translation lead to loss or addition of extracted feature points leading to a different Delaunay tessellation. The difference in the tessellation leads to synchronization errors as it is more difficult to estimate the rotation, scaling and translation (RST) attack. Our proposed scheme uses averaging of triangles to estimate the exact nature of attack on the image in order to restore it to the position of the original image. This ensures that the watermark can still be extracted even after losing a number of feature points.

This paper is organized as follows. Section (II) describes the extraction of feature points. Section (III) describes Delaunay triangulation. Section (IV) describes the watermark embedding process, the watermark recovery procedure and the proposed scheme presented. Section (V) presents the experimental results and the conclusion is given in Section (VI).

II. FEATURE-POINTS EXTRACTION

The Harris corner detector measures the local changes in the image by shifting small windows in different directions.

The auto-correlation function used in determined the changes is defined as [7]

\[ C(x,y) = [\Delta x \Delta y] C(x,y) [\Delta x \Delta y]^T \]  

Where \( (\Delta x, \Delta y) \) is the small amount of shift of windows and matrix \( C(x,y) \) is given by

\[ C(x,y) = \begin{bmatrix} \sum_w I_x(x_i,y_i)^2 & \sum_w I_x(x_i,y_i)I_y(x_i,y_i) \\ \sum_w I_y(x_i,y_i)I_x(x_i,y_i) & \sum_w I_y(x_i,y_i)^2 \end{bmatrix} \]  

Where \( w \) is the small square Gaussian window whose dimensions are of odd number of pixels centered at \((x,y)\) and \( I_x(x,y) \) and \( I_y(x,y) \) are partial derivatives of the image function \( I(x,y) \) in \( x \) and \( y \) axes respectively.

The eigenvalues of the matrix \( C(x,y) \) are used to determine if the windowed image region has any feature-points. A threshold is then set such that when both the eigenvalues exceed it, then the windowed region is considered to have a feature-point.

The feature-points are then employed in generating the Delaunay triangulation which is used in the synchronisation process [3], [5].

III. DELAUNAY TRIANGULATION

The Delaunay triangulation employs the concept of empty circumcircle, that is, for any three points to form a Delaunay triangle, all of them have to lie on a circumcircle with no point lying inside it.

It has the property of giving unique tessellation for a given set of feature-points except when four points lie on the circumcircle. Due to this uniqueness property and the tessellation consists of triangles only which are simpler to restore make Delaunay triangulation the most suitable for synchronisation of distorted images [3].

IV. RESTORATION OF ATTACKED IMAGES

Before attempting to extract the embedded watermark from a geometrically distorted image, it is first restored back to its original position. Estimation of the nature and magnitude of the attack is done and then a reversal of the attack implemented. The process of synchronisation can be done by either template matching or feature-point based techniques [4]-[6] and [8].

Pereira and Pun [8] proposed a technique where log-polar maps are used to embed a geometric attack resilient template in the DFT domain which is used to synchronise the attacked image. This technique has the advantage of being robust to both local and global attacks and the disadvantage of being computationally complex.

Tang and Hang [4] uses Mexican-hat wavelet filter in the extraction. The watermark is embedded in local Voronoi regions which employ the feature-points as their centroids. They employed scale-space theory in the extraction of feature-point extraction to improve the robustness of the extracted feature-points. However this technique was not sufficiently robust to most rotation and scaling attacks.

Qi and Qi [5] propose improving the robustness of the of the extracted feature points by using a circular window in the Harris corner detector. This technique lacked sufficient robustness to angles that led to formation of substantial dark regions around the image and did not improve robustness to cropping attacks.

Bas et al [6] use a pre-filter blur to reduce the sensitivity of the Harris corner detector and a circular neighbourhood to obtain a homogenous distribution of feature points. Since this watermarking technique is content based, it was found to be quite robust to most attacks. However the embedding and extracting algorithms exhibited substantial increment in complexity as the watermark is embedded into a set of disjoint triangles which are generated by triangulating a pseudo-random set of points.

V. PROPOSED METHOD

The proposed algorithm uses VQ, DCT and Spread-spectrum techniques during the watermark embedding process and incorporates Delaunay triangulation in the extraction process. The embedding procedure can be summarized as follows. The image is decomposed into
its component images and the embedding component chosen depending on the colour space. The LBG vector quantisation algorithm is employed in the design and subsequent generation of the codebook of the encoded image, the codebook is again decomposed into 8-by-8 pixel blocks [8]-[11] and the DCT coefficients of each block obtained. The DCT coefficients of the watermarks are then embedded into the coefficients of the codebook in a spread-spectrum format.

The watermark extraction process can be summarized as follows. Delaunay triangulation of the feature points of the image is used to resynchronize the image to the position of the original image. After resynchronization, the codebook is obtained and the DCT coefficients obtained and the watermark extracted from the embedded positions.

### A. Embedding Procedure

**Step 1.** Decompose the cover image into component images and select the embedding component.

**Step 2.** Perform VQ encoding of the image using the LBG algorithm to construct a codebook of the image with a codeword of size $8 \times 8$ (64).

**Step 3.** Decompose the codebook into $8 \times 8$ non-overlapping blocks and compute a block-based DCT of the codebook and then select the coefficients of the mid-frequency sub-band.

**Step 4.** Generate a pseudo-random sequence and select the value of $\beta$ which determines the trade-off between visual imperceptibility and robustness.

**Step 5.** Embed the watermark into the selected mid-frequency sub-band using the pseudo-random (PN) sequence to determine the coefficients where the watermark is embedded as follows:

$$I^w(u,v) = I(u,v) + \beta W(u,v)$$  (3)

Where, $I(u,v), I'(u,v)$ and $W(u,v)$ are the original, watermarked and watermark DCT coefficients of the codebook respectively.

**Step 6.** Compute the inverse DCT of the watermarked codebook, perform VQ decoding and then concatenate the component images to obtain the watermarked colour image.

The value of $\beta$ varied between 0.1 and 0.5 depending on the pixel texture of the image [10]. It was observed that the generally smooth textured images a higher value of $\beta$ could be used with less visual quality distortion than the highly textured images.

### B. Watermark Recovery Procedure

The process of watermark recovery from the image is given in this subsection as follows:

**Step 1.** Extract the feature-points of the colour image using the Harris corner detector then obtain

the Delaunay triangulation of the feature-points obtained.

**Step 2.** Select several target triangles and corresponding probe triangles from the Delaunay of the original and attacked image respectively.

**Step 3.** Using the triangles estimate the amount of distortion that each probe triangle has been subjected to by comparing the orientation angles, sizes and positions to the corresponding target triangle.

**Step 4.** Compute the mean of the distortions of all triangles and use the averages as the rotation factor (RF), scaling factor (SF) and translation factor (TF) which are then used to restore the image to its original form. The factors are obtained using the formula given in equation 4.

$$F_{avg} = \frac{F_1 + F_2 + \cdots + F_N}{N}$$  (4)

Where $F_{avg}$ is the mean of the distortions obtained from the probe triangles, $F_N$ is the distortion obtained from the $N$th probe triangle.

**Step 5.** Decompose the resynchronised colour image into its component images and then select the embedding component.

**Step 6.** Obtain the codebook of the embedding component image and compute its DCT.

**Step 7.** Select the mid-band frequencies and using the same PN sequence used in the embedding process check for the presence or absence of the embedded watermark using correlating the coefficients of the mid-band frequencies and the PN sequence.

**Step 8.** Reconstruct the watermark using the extracted watermark bits.

### VI. EXPERIMENTAL RESULTS

Several experiments were performed in order to test the effectiveness of our proposed watermarking technique. The technique was tested for visual perceptibility and robustness to a variety of attacks and the results given in the following subsections.

#### A. Visual Perceptibility

The watermarked images were assessed for visual distortion using Peak Signal-to-Noise Ratio (PSNR). The Couple test image in figure 1 showed no visible visual degradation in quality at a PSNR value of 43.3dB. Similar observations were noted for other test images such as Peppers, Baboon and Lena.

#### B. Robustness to Attacks

The watermarked image was then tested for robustness to signal processing attacks and geometric distortion attacks. The watermark was then extracted after restorations where necessary and the quality of the
Appendix C: Publication 2

watermark computer using a Normalized cross-correlation (NC) factor.

Fig. 1 The (a) original and (b) watermarked images at a PSNR value of 43.3 dB

The first attack to be simulated was the JPEG compression attack where the image was compressed to various quality factors before attempting to extract the watermark. Figure 2 shows the peppers test image after being subjected to JPEG compression at 30% QF and the extracted watermark.

Fig. 2 The (a) JPEG compressed image at QF = 30% and (b) extracted watermark at NC of 0.99

The images were then scaled from 25% to 150% of the original image size. Fig. 5 shows the tessellation, probe triangle and the watermark recovered from the attacked baboon test image at an SF of 1.5.

Translation attacks for various TF were also simulated and the watermark was recovered. Figure 7 shows Baboon image translated by 15 pixels to the right with its probe triangles and the extracted watermark. Figure 8 illustrates an image subjected to shear attack and the watermark which was extracted from it.

Finally a variety combinations of RST attacks were simulated to test the robustness of the algorithm to combinations of attacks and figure 9 shows Peppers image subjected to RST attack with RF of 35°, SF of 0.75 and TF of (10,0) and the extracted watermark.

Fig. 3 The (a) probe triangle and (b) target triangle after a huge rotation distortion RF = 29°.

(a)
(b)

Fig. 4 Watermark embedded in cameraman image (a) original message and (b) extracted message after restoring rotation attack

(a)
(b)

Fig. 5 (a) Delaunay triangulation and (b) the probe at SF = 1.5

(a)
(b)

Fig. 6 The recovered message from figure 5 above at NC = 0.96.
Appendix C: Publication 2

![Baboon image](image1)

**Fig. 7** The *Baboon* image after translation attack with TF = 15 and the extracted message at NC = 0.96

![Peppers image](image2)

**Fig. 8** The (a) *Peppers* image after a shear attack of 8% and the extracted message at NC = 0.92 and the (b) extracted watermark.

![Peppers image](image3)

**Fig. 9** The (a) *Peppers* image after a combination of RST attacks and (b) extracted watermark at NC of 0.88.

Computer simulation experiments were performed on the images to test the robustness of the watermarking algorithm to filtering, JPEG compression and histogram equalization attacks and a summary of the results given in table I.

<table>
<thead>
<tr>
<th>Table I.</th>
<th>NC OF VARIOUS ATTACKS ON VARIOUS IMAGES (OUR PROPOSED SCHEME VS TANG’S [3])</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Couple</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
</tr>
<tr>
<td>JPEG (30%)</td>
<td>0.99</td>
</tr>
<tr>
<td>JPEG (50%)</td>
<td>0.99</td>
</tr>
<tr>
<td>Avg. filt.</td>
<td>0.99</td>
</tr>
<tr>
<td>Unsharp filt.</td>
<td>0.99</td>
</tr>
<tr>
<td>Histo. equal.</td>
<td>0.99</td>
</tr>
<tr>
<td>Cropping 0.2</td>
<td>0.91</td>
</tr>
<tr>
<td>Rotation (29°)</td>
<td>0.99</td>
</tr>
<tr>
<td>Scaling (1.5)</td>
<td>0.96</td>
</tr>
<tr>
<td>Translation(15,0)</td>
<td>0.97</td>
</tr>
<tr>
<td>Shear 8%</td>
<td>0.94</td>
</tr>
<tr>
<td>RST (35°;0.75;10,0)</td>
<td>0.85</td>
</tr>
</tbody>
</table>

The proposed technique performed better than the other schemes as seen in table I.

VII. CONCLUSION

In this paper, an image watermarking technique resistant to affine geometric attacks has been presented. The experimental results demonstrate that watermarks can be extracted from images that have been subjected to severe geometric distortion. The proposed algorithm offers improved robustness to geometric attacks with minimal increase of computational complexity.

VIII. ACKNOWLEDGEMENT

The authors would like to thank Kenya Education Network for the financial grant provided to conduct this research.

REFERENCES


Appendix C: Publication 2


